IDA

INSTITUTE FOR DEFENSE ANALYSES

Security Criteria for Distributed Systems: Functional Requirements

Terry Mayfield, Task Leader

Virgil D. Gligor
Janet A. Cugini
John M. Boone
Robert W. Dobry

September 1995

IDA Paper P-3159

Log: H 95-047505

This work was conducted under contract DASW01 94 C 0054, Task T-AA5-962, for the National Security Agency. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

PREFACE

This document was prepared by the Institute for Defense Analyses (IDA) under the task order, Federal Criteria Development, and fulfills the objective of extending the Federal Criteria to support distributed operating systems. The study was sponsored by the National Security Agency (NSA) with the joint involvement of the National Institute of Standards and Technology. The study was initiated as a separate, parallel effort to that of developing the international Common Criteria, with the intent of making this study's material available at an appropriate time for ultimate inclusion into the Common Criteria.

The authors are greatly indebted to critical reviews, contributions, and guidance provided by a distinguished review panel consisting of security, cryptographic support, communications, and distributed systems experts. The panel members were Kenneth Birman, Department of Computer Science, Cornell University; Whitfield Diffie, Sun Microsystems, Inc.; Stephen Kent, Bolt Beranek and Newman, Inc.; Butler Lampson, Microsoft Inc., formerly with Digital Equipment Corporation's Systems Research Center; John Linn, OpenVision Technologies; B. Clifford Neuman, University of Southern California - Information Sciences Institute; Robert Morris, NSA; and Peter Weinberger, AT&T Bell Laboratories. The authors are also grateful to Grant Wagner, NSA, and Ron Ross, IDA, for additional reviews and comments.

The inputs of all reviewers have been invaluable to the formulation and completion of this work; however, their participation in the reviews does not constitute endorsement of the results. The authors retain full responsibility for the results and believe that this final draft remains faithful to the constructive intent of the panel members and other reviewers.

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

Background

This document presents functional security requirements that can be used to assemble evaluation criteria for the security features of trusted distributed computing systems. Because of continuous advances in computer system technology, distributed systems have emerged as an important area for the DoD. The requirements presented in this report build on the established technical base of security criteria, and towards the establishment of new, internationally accepted criteria for this technology area. These requirements are the first to specifically address criteria associated with trusted distributed systems.

This task was conducted by the Institute for Defense Analyses (IDA) for the National Security Agency (NSA). IDA has participated in the formulation of security criteria for a number of years and was a major contributor to the Federal Criteria[1] which provided an update to existing, DoD-related security criteria (i.e., the "Orange Book").[2] The Federal Criteria effort produced broader, generic criteria that included the security concerns of non-classified Federal computing environments.

Upon the completion of version 1 of the Federal Criteria, IDA was tasked to undertake a separate parallel effort to extend those criteria by defining requirements for distributed systems while NSA, the National Institute of Standards and Technology (NIST), and representatives from other nations proceeded to develop the harmonized Common Criteria.[3] A goal of IDA's distributed systems tasking was to enable the study and development of new distributed systems security requirements without interfering with the harmonization of various sets of existing criteria. This set of distributed criteria, while separately pub-

---

[1] U. S. Department of Commerce, National Institute of Standards and Technology (NIST), and the National Security Agency. December 1992. *Federal Criteria for Information Technology Security.* Volumes I and II. Version 1 (Draft). Gaithersburg, MD: NIST.

[2] U. S. Department of Defense. 1985. *Trusted Computer System Evaluation Criteria.* DoD 5200.28-STD. Washington, DC: U. S. Government Printing Office.

[3] Government of Canada, Communications Security Establishment. October 24, 1994. *Common Criteria for Information Technology Security Evaluations, Rationale, Parts 1, 2, and 3. Version 0.9.* CCEB-94/089 (Draft). Ottawa, Canada: Canadian System Security Centre.

lished, is ready to be incorporated into the Common Criteria which is nearing its final development phase.

Scope

The focus of the security requirements presented in this report is on the design, implementation, and operation of trusted distributed operating systems. The view represented in this report is that any trusted distributed system consists of a set of Trusted Computing Bases interconnected by trusted channels subject to interconnection policies, or constraints, placed on one or several security perimeters. A detailed rationale for this view of a distributed-system product is provided in the National Research Council's report, *Computers at Risk.*[4]

What is presented herein is not intended to stand alone: these requirements rely on continuing work (i.e., the Common Criteria) to provide a process and infrastructure by which they can be assembled into specific evaluation criteria and subsequently applied (e.g., in the evaluation of a trusted distributed system). These requirements apply only to the *functional* security requirements of distributed systems. Functional security requrirements relate to mechanisms implementing system and information protection. The development of additional *assurance* requirements is needed to have a complete set of requirements for trusted system evaluation criteria. Assurance requirements are those that affect the "trust" or confidence one has in the design, construction, and operation of a given protection feature or mechanism. The development of a set of assurance requirements will occur via the Common Criteria working group.

The requirements presented in this report borrow heavily from the strong foundational work that resulted in the draft security criteria known as the Federal Criteria. The technical content and focus of those criteria were adapted to incorporate the area of distributed computer systems. Part of this adaptation was to make the presentation of requirements as modular as possible, with the intention of making them more usable and adaptable, thus extending their life with the emergence of new technologies.

Requirements Classes

---

[4] National Research Council. 1991. *Computers at Risk: Safe Computing in the Information Age.* Washington, DC: National Academy Press.

This document presents requirements for 12 classes of security features, shown in Table ES-1. Specific functional security requirements have been developed in each of these

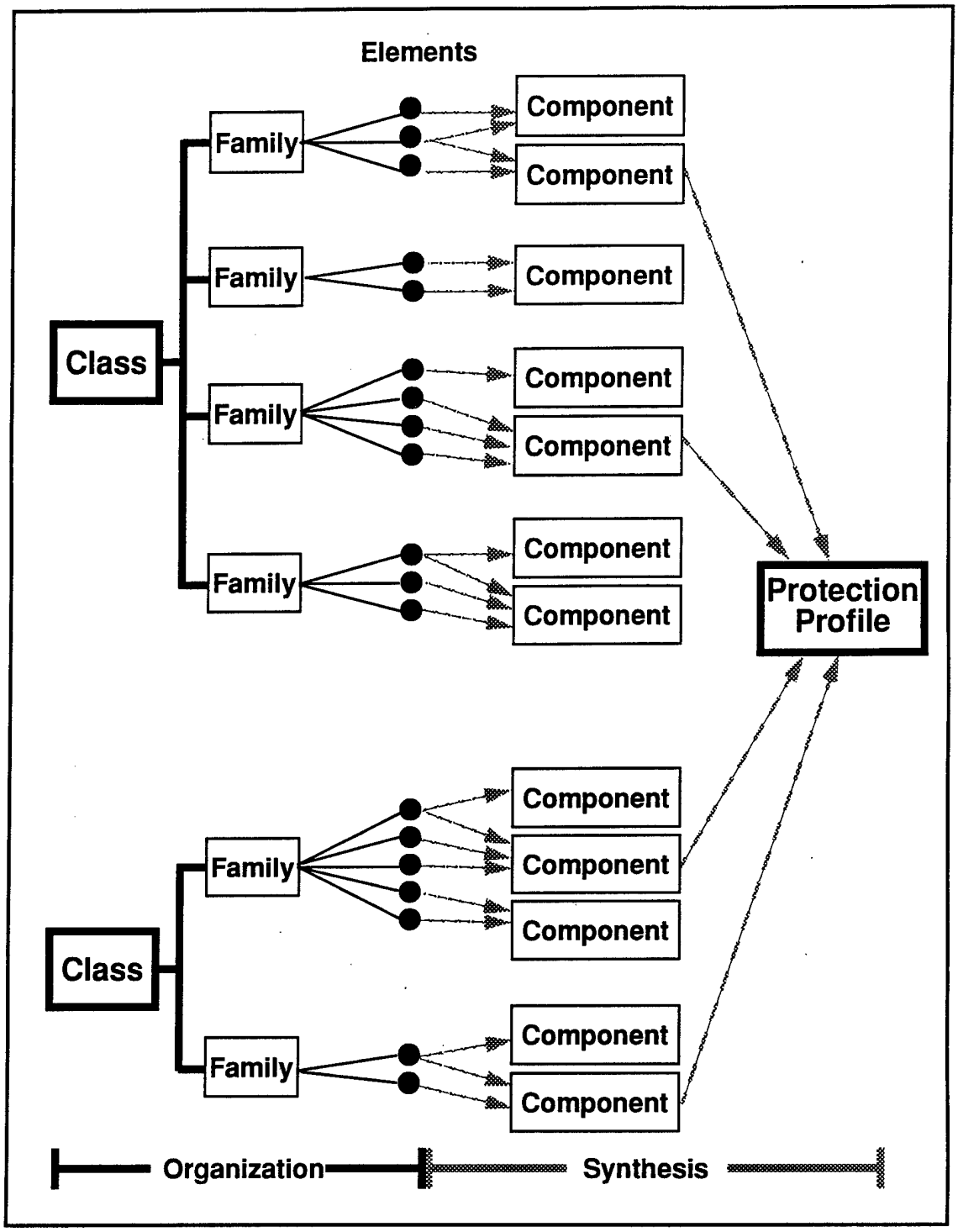
Table ES-1. Functional Requirements Classes

Section	Requirements Classes
A	Trusted Computing Base
B	Identification and Authentication
C	System Entry
D	Trusted Path
E	Data Confidentiality
F	Data Integrity
G	Cryptographic Support
H	Access Control
I	Covert Channel Countermeasures
J	Audit
K	Availability (TBD)[a]
L	Security Management

a. Due to a weaker technical foundation for this class of functional requirements, no requirements for Availability were attempted for this study. The inclusion of this (empty) class simply acknowledges this as an increasingly important area.

classes. The document builds on the established framework of traditional computer security areas such as Trusted Computing Base protection and Access Controls. The requirements interpret these traditional areas in the context of distributed systems and extend the established base, where necessary, within areas that are of particular interest in distributed systems (e.g., Cryptographic Support, Data Integrity, and Data Confidentiality).

In addition to providing actual requirements, this report extends previous efforts at improving security criteria methodology. Two terms are used, *organization* and *synthesis*, that describe the taxonomic structuring of how the distributed systems security requirements are to be used in practice. These terms were first defined in this context by the Federal Criteria, and the meanings here are identical to the earlier usage. Figure ES-1 illustrates the complementary structuring of organization and synthesis.



Adapted from Figure 4.1 of *Common Criteria for Information Technology Security Evaluations, Rationale, Parts 1, 2, and 3. Version 0.9.* CCEB-94/089 (Draft). Ottawa, Canada: Canadian System Security Centre. Government of Canada, Communications Security Establishment.

Figure ES-1. Requirements Organization and Synthesis

Organization refers to the top-down grouping of requirements into logical domains in a class-family-element hierarchy. Organization provides a means of requirements structuring, naming, evolution, and extensibility. *Synthesis* refers to the bottom-up grouping of requirements, forming an element-component-profile hierarchy (with protection profiles at the top). Synthesis provides a means of requirement groupings that are useful in the specification, analysis, and evaluation of secure systems and system components.

The highest organizational structure of the distributed systems security requirements is termed a *class*. Each class refers to either a traditional, generic computer security area (e.g., access control) or to a particular area that is relevant to distributed systems. The entire set of requirements classes forms a taxonomy of functional security requirements. Each class can be decomposed into one or (usually) more "families." A *family* is a particular set of requirements that relate to the overall class and address a logical class of threats. For instance, there is an Audit Management family under the Audit class that specifies requirements related to operation of an audit subsystem in a distributed system.

The individual requirements, termed *elements*, are organized under families. Elements are highly modular, with individual elements having a distinct reference (i.e., a name). Elements are intended as the smallest coherent statement of a security requirement, although elements are not generally limited to a singular statement. The modularity of an element provides a convenient artifact for tying together information related to a particular requirement (e.g., any dependencies and/or parameters the requirement may have).

The element notation developed in this report allows for the evolution of requirements into related, yet distinct, elements. Related elements are often "stronger" or a more rigorous adaptation of the base element. This abstraction provides a flexibility in specification that was notably absent from the Orange Book.

Elements are also the basic structural artifact in the synthesis of security specifications. Element specifications are grouped into *components*. A component binds a set of elements into a useful specification that can be applied to actual implementations and evaluations of systems and/or system components (e.g., an audit subsystem). The central notion at this level of synthesis is that many useful components can be developed through the selection of exactly the requirements (i.e., elements) desired. By selectively grouping stronger elements into different components, two goals are attained. First, security specifications can be more precisely identified, avoiding the problem of "over specified" security requirements. Second, a series of related components can be built up, allowing for a natural structuring of components into an arbitrary classification scheme.

Components can be grouped into the highest-level structural artifact of synthesis, a (protection) *profile*. Profiles will be used as computer security product evaluation criteria. Again, the degree of flexibility provided by the modularity of the requirements is preserved: different component specifications can be "mixed and matched" to exactly specify the intended security features of a system, and new profiles can be evolved (as new elements and components are evolved). Such ease of evolution is intended to better match the pace of technology advancement, a serious problem in previous criteria.

It is expected that these requirements can be put to immediate use in the specification, design, implementation, and evaluation of secure, distributed computing systems. In addition, the criteria foundation presented here can be easily evolved, allowing it to grow as technology advances. Adapting a style first set forth in the Federal Criteria, extensibility is built into the requirement-naming conventions, and the requirements are presented in a modularized format. The modularity of requirements helps to accommodate the technical development of new classes, families, elements, components, and protection profiles, and helps avoid some of the extensibility drawbacks of earlier computer security criteria.

PART 1. FUNCTIONAL REQUIREMENTS

1. OVERVIEW

This report presents a set of functional security requirements that can be used to assemble evaluation-oriented criteria for distributed systems and components. These requirements capture necessary security characteristics of distributed systems, enable the definition of specific protection profiles (evaluation criteria) for trusted distributed systems that can be used in various threat environments, and allow for protection profile extension and refinement which may be needed as technology evolves, threats change, and experience is gained in specifying and evaluating distributed systems.

Part 1 presents information that is intended to help the reader understand and use the distributed systems functional requirements, which are presented in Part 2. Chapter 1 presents the authors' conceptual view of trusted distributed systems, and the scope, structure, operations, and dependencies of the requirements. Chapter 2 presents the various naming conventions used in the criteria.

Part 2 presents the distributed systems functional security requirements developed for this report. The requirements are organized into sections, each covering a separate class of security requirements. There are 12 classes of functional security requirements: Trusted Computing Base, Identification and Authentication, System Entry, Trusted Path, Data Confidentiality, Data Integrity, Cryptographic Support, Access Control, Covert Channel Countermeasures, Audit, Availability, and Security Management.

This task was conducted by the Institute for Defense Analyses (IDA) for the National Security Agency (NSA) as follow-on work for the Federal Criteria[1] which provided an update to existing Department of Defense (DoD) security criteria (i.e., the "Orange Book").[2] The initial Federal Criteria effort produced broader, generic criteria for stand-

---

[1] U. S. Department of Commerce, National Institute of Standards and Technology (NIST), and the National Security Agency. December 1992. *Federal Criteria for Information Technology Security.* Volumes I and II. Version 1 (Draft). Gaithersburg, MD: NIST.

[2] U. S. Department of Defense. 1985. *Trusted Computer System Evaluation Criteria.* DoD 5200.28-STD. Washington, DC: U. S. Government Printing Office.

alone computing systems that included the security concerns of non-classified Federal computing environments.

IDA has participated in the formulation of security criteria for a number of years and was a major contributor to the Federal Criteria. Upon completion of version 1 of the Federal Criteria, IDA was tasked to undertake a separate, parallel effort to extend that criteria by defining requirements for distributed systems while NSA, the National Institute of Standards and Technology (NIST), and representatives from other nations undertook the development of the harmonized Common Criteria. This separate tasking was to enable the study and development of new distributed systems requirements without interfering with the harmonization efforts. This set of functional security requirements for trusted distributed systems, while separately published, is ready to be incorporated into the Common Criteria[3] which is nearing its final development phase.

1.1 TRUSTED DISTRIBUTED COMPUTER SYSTEMS

The view represented in this report is that any trusted distributed system consists of a set of Trusted Computing Bases (TCBs) interconnected by trusted channels subject to interconnection policies, or constraints, placed on one or several security perimeters. A detailed rationale for this view of a distributed-system product is provided in the National Research Council's report, *Computers at Risk.*[4]

The *TCB* of a secure information processing system, not just a typical computer system, consists of the hardware, firmware, and software code and data structures responsible for enforcing the system's protection functions.

A *channel* is an information path by which two or more subjects can communicate. A trusted channel provides data confidentiality, which enables the sender to know who can read a message it sent; data integrity, which enables a receiver to know that the message it received is unmodified and, therefore, also enables the receiver to know who originally created the message; authentication, which enables both the sender or the receiver to find out who is at the other end of a channel; and availability, which enables the sender to know that his message will be received by the intended receiver.

---

[3] Government of Canada, Communications Security Establishment. October 24, 1994. *Common Criteria for Information Technology Security Evaluations, Rationale, Parts 1, 2, and 3. Version 0.9.* CCEB-94/089 (Draft). Ottawa, Canada: Canadian System Security Centre.

[4] National Research Council. 1991. *Computers at Risk: Safe Computing in the Information Age.* Washington, DC: National Academy Press.

A *security perimeter* represents a partition of a distributed-system product that delimits both the scope of the administrative control over the product and application resources (e.g., hosts, communication gateways) as well as the scope of security policies enforced unilaterally by a single, centralized administrative organization.

Interconnection policies, or constraints, consist of a set of rules that define whether trusted channels may be established between the TCBs of a security perimeter and among different security perimeters, and the types of those trusted channels (e.g., confidentiality only, integrity and availability, authentication only).

1.2 SCOPE

The scope of these functional security requirements covers the range of requirement applicability (i.e., the types of distributed systems we intend to address), the parts of a distributed system to which the requirements apply, and the specification target—those distributed-system entities that are subject to the stated requirements (i.e., security functions vs. system services).

Range of Applicability. The functional requirements presented here refer to the operating systems of distributed-system products. They do not address explicitly or completely the requirements of application security for distributed systems. For example, security requirements for trusted notarization services, document signature verification, electronic cash, secure teleconferencing, or secure elections in a distributed system are not explicitly addressed. Security requirements for those applications are considered outside the scope of this document. However, the requirements of this document are intended to be consistent with and support the requirements of such applications.

Relevant Parts. The functional requirements presented here are based on the premise that the host TCBs and trusted channels are the only parts of a trusted operating system for distributed-system products that need to be analyzed and evaluated to determine its protection characteristics. This premise is valid because TCBs and trusted channels implement security perimeters and interconnection constraints. This premise is also valid for a wide variety of distributed systems ranging from communication networks, where different types of channels and interconnection policies can be selected by users and applications, to integrated distributed systems, where a uniform set of security policies is enforced by a common security infrastructure.

Specification Target. The individual functional requirements presented here refer to security functions rather than to distributed system services. This choice is made for the

following three reasons. First, most system services, including directory, file, input/output, and inter-process communication and synchronization services, share the same, or very similar, security requirements. Therefore, per-service requirement specification would lead to significant redundancy. This is the primary reason why, unlike existing communication standards, existing security criteria have typically not chosen a service-oriented requirements specification approach.

Second, a service-based requirements specification would inevitably contribute to the ongoing "layer wars" in the communication network area, since many seemingly similar security requirements appear in several layers of communication protocols. Controversy as to which service and layer are more suitable for a specific security function can be avoided by specifying generic function-oriented components and individual requirements that can be used, instantiated, and refined in different service and layer contexts, as the need arises.

Third, requirement specifications for security functions, rather than for system services, appear to be generally accepted by the security community. The choice of specifying individual requirements and components on a security-function basis is more natural, given the importance of the need to integrate requirements for centralized-system products with those for distributed-system products, and the pervasiveness of existing centralized-system criteria, all of which have a security-function orientation.

1.3 FUNCTIONAL REQUIREMENTS STRUCTURE

The structure of the presented requirements is intended to satisfy three independent goals: naming, which provides ease of requirement identification and location; linkage, which provides an ability to assemble and synthesize requirements into coherent sets similar in structure with those of traditional standards (see *Other Security Criteria* section of the Bibliography); and compatibility, to the largest possible extent, with the current structure of the Common Criteria for information security technology.

To satisfy these goals, we use both an organization taxonomy for requirement naming, and a functional taxonomy for requirement synthesis. The two taxonomies differ primarily because they serve different purposes. They also differ because of requirement rating, which is reflected more in requirement naming and less in synthesis. That is, a basic requirement may have several versions that need to be named to reflect rating differences resulting from variations in the scope of the requirement application, the granularity of the requirement application, the coverage of the security features necessary to satisfy the

requirement, and the strength of the requirement.[5] In contrast, requirement synthesis selects a single named version of a given type of requirement. The two taxonomies and their relationship are illustrated in Figure 1.

Organization Taxonomy. The most basic requirement unit is called the *element.* The element is intended to be the smallest requirement unit that can be separately analyzed and evaluated. An element may have several versions, each reflecting an element rating. A group of independently identified elements that share a single purpose forms a *family.* A total ordering among some, but not necessarily all, of the elements of the same family results from element rating (e.g., only some of the elements of the Data Integrity family are rated). One or more families form a *class.* A class may denote the set of requirements that characterize a security policy (e.g., Identification and Authentication, Audit, Access Control, Data Confidentiality, Data Integrity), or may denote salient functional or mechanism features (e.g., TCB functions). A class is intended to provide a requirement grouping reminiscent of that used in traditional security-requirement standards (see *Other Security Criteria* section of the Bibliography). One may also provide a hierarchical structure for classes; however, we avoid using such additional class structure in this report to simplify requirement identification and location.

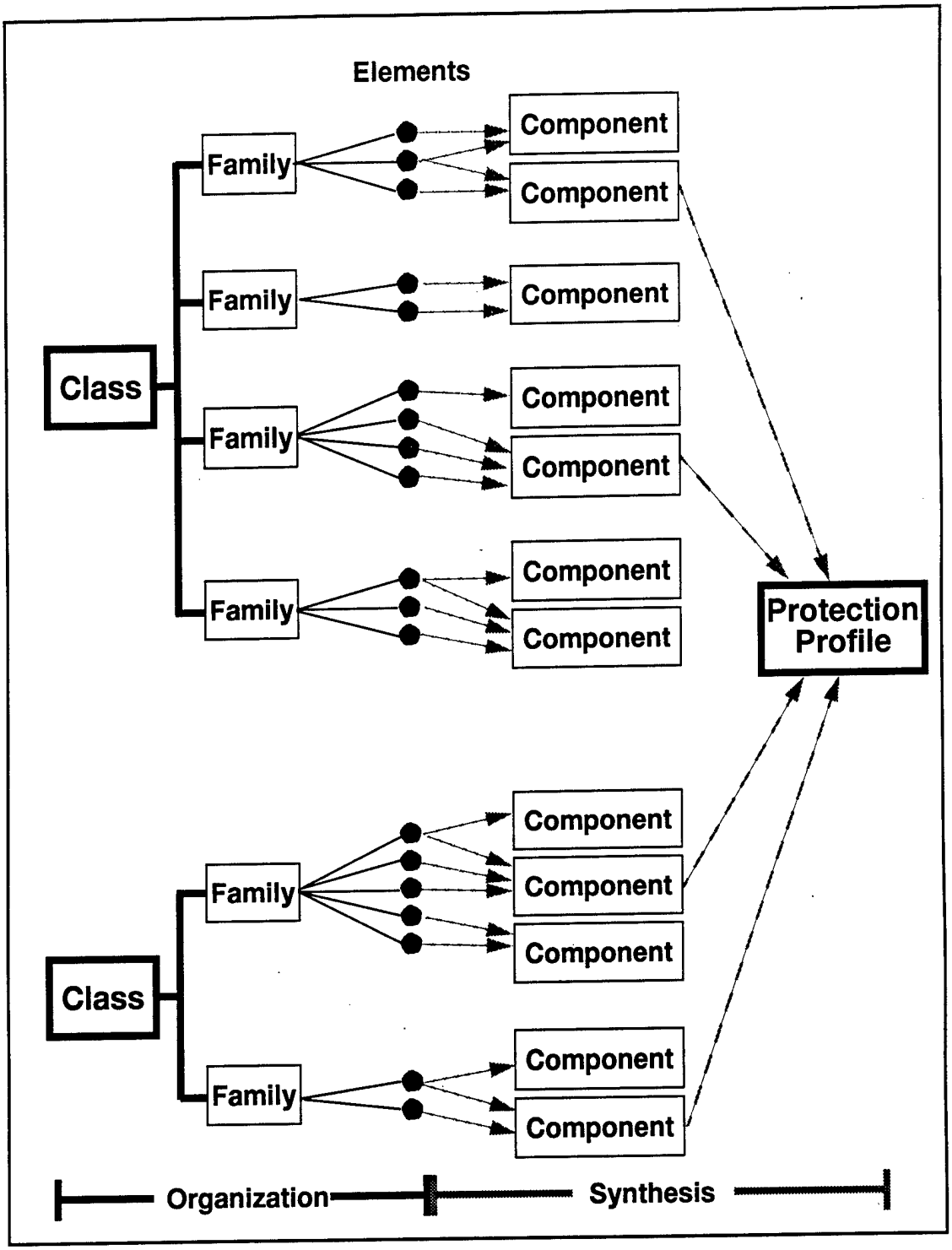
Synthesis Taxonomy. Elements of a family are selected and assembled into *components.* Component assembly requires that each included element represents a single version of a requirement. The selection of specific elements (i.e., requirement versions) and the number of elements assembled form the basis for component rating in a similar manner as that of individual requirement (i.e., element) rating. We note that, as is the case with element rating, component rating can lead to a partial order among the components of a family.

Components are assembled into protection profiles, or simply *profiles.* A profile consists of a set of requirements that characterize the set of related security policies and mechanisms needed to counter a set of threats and address a set of security vulnerabilities in the environment(s) of system use.[6] As such, the profile represents the output of the requirement synthesis process. Its structure and content are similar to that of the evaluation classes of traditional trusted system evaluation criteria standards.

---

[5] U. S. Department of Commerce, National Institute of Standards and Technology (NIST), and the National Security Agency. December 1992. *Federal Criteria for Information Technology Security.* Volumes I and II. Version 1. Version 1 (Draft). Gaithersburg, MD: NIST

[6] Ibid.



Adapted from Figure 4.1 of *Common Criteria for Information Technology Security Evaluations, Rationale, Parts 1, 2, and 3. Version 0.9.* CCEB-94/089 (Draft). Ottawa, Canada: Canadian System Security Centre.Government of Canada, Communications Security Establishment.

Figure 1. Requirements Organization and Synthesis

1.4 FUNCTIONAL REQUIREMENTS DEPENDENCIES AND OPERATIONS

The assembly of elements into components, and of components into profiles, requires that the dependencies among elements and components be identified. Dependencies among elements arise because security functions that implement an element depend on security functions implementing other elements, or because security functions that implement different elements must support the same policy either individually or collectively. Thus, a distinction is made between the "uses" and "policy" dependencies among elements.[7] Element dependencies are important in component and profile assembly because they help identify what must be included in a component. The elements included in this report only reflect direct element dependencies. Therefore, because dependencies are transitive, the assembly of components must discover the transitive closure of all elements.

Other dependencies that arise in the process of assembling components into protection profiles are not included in this report. Also not included are examples of operations that can be performed on individual elements to form components (e.g., assignment, refinement, augmentation).[8]

---

[7] Ibid.

[8] Ibid.

2. REQUIREMENTS ORGANIZATION AND CONVENTIONS

This chapter presents the overall organization of the distributed systems functional security requirements presented in Part 2 of this report. In addition, naming conventions used in structuring the requirements are described. The information presented in this chapter is intended to be used as reference material when using the criteria.

2.1 REQUIREMENTS ORGANIZATION

As discussed previously, requirements are expressed as elements. The elements and families of elements presented in this document are divided into classes as shown in Table 1 on pages 12-13. The four groups of requirement classes reflect the principal security concerns of a distributed system, namely (1) the protection of the TCB of each host, (2) channel security functions and policies, (3) access control, audit, and availability policies, and (4) security management, which supports all three previous groups of requirement classes.

There is a separate document section for each class of elements currently defined for the distributed criteria. For each class of elements, we provide in the criteria a brief description of the families included in the class, the threats intended to be countered, a list of elements, and several examples of components illustrating the use of elements in component composition. Following the description section of each class are separate subsections for each family within the class. Each family section is divided into two parts: one composed of a list of elements belonging to that family, and the other composed of several examples of components illustrating the use of the elements in component synthesis.

It should be noted both the list of element families and the list of elements within a family can be extended as technology matures. Similarly, readers should also view individual components as modifiable (using the operations specified in the Federal Criteria[9]) and the list of components as extensible. This approach provides a common, relatively controlled security requirements specification language with the flexibility that allows such requirements to evolve as needed to meet a wide variety of possible protection needs.

---

[9] Ibid, Chapter 7.

Table 1. Functional Requirements Classes and Families

Requirements Class	Class Abbr.	Requirements Family	Family Abbr.	Section (Page)
Group 1				
Trusted Computing Base	TCB	—	—	A (p. 19)
		Reference Mediation	RM	A.1 (p. 25)
		Logical TCB Protection	LP	A.2 (p. 29)
		Physical TCB Protection	PP	A.3 (p. 33)
		TCB Self-Checking	SC	A.4 (p. 37)
		TCB Start-Up and Recovery	SR	A.5 (p. 41)
		TCB Privileged Operation	PO	A.6 (p. 45)
		TCB Ease-of-Use	EU	A.7 (p. 51)
Group 2				
Identification and Authentication	IA	—	—	B (p. 55)
		Identification	IAI	B.1 (p. 57)
		Channel Authentication	CA	B.2 (p. 61)
		User Authentication	UA	B.3 (p. 69)
		Inter-Realm Authentication	IRA	B.4 (p. 77)
		Authentication Policy	IAP	B.5 (p. 81)
System Entry	SE	—	—	C (p. 83)
		Distributed System Entry	DSE	C.1 (p. 85)
Trusted Path	TP	—	—	D (p. 91)
		Distributed Trusted Path	DTP	D.1 (p. 93)
Data Confidentiality	DC	—	—	E (p. 97)
		Data Confidentiality Functions	DCF	E.1 (p. 99)
		Data Confidentiality Policy	DCP	E.2 (p. 107)
Data Integrity	DI	—	—	F (p. 115)
		Data Integrity Functions	DIF	F.1 (p. 117)
		Data Integrity Policy	DIP	F.2 (p. 129)

Table 1. Functional Requirements Classes and Families (Continued)

Requirements Class	Class Abbr.	Requirements Family	Family Abbr.	Section (Page)
Cryptographic Support	CR	—	—	G (p. 135)
		Secure Cryptographic Function	SCF	G.1 (p. 137)
		Cryptographic Domain Protection	CDP	G.2 (p. 141)
		Secure Key Management	SKM	G.3 (p. 145)
Group 3				
Access Control	AC	—	—	H (p. 155)
		Definition of Access Control Attributes	ACA	H.1 (p. 157)
		Authorization of Subject Access to Objects	SAO	H.2 (p. 163)
		Administration of Access Control Attributes	AA	H.3 (p. 171)
Covert Channel Countermeasures	CC	—	—	I (p. 175)
		Covert Channel Handling	CCH	I.1 (p. 177)
Audit	AU	—	—	J (p. 181)
		Audit Protection	AP	J.1 (p. 183)
		Auditable Events	AE	J.2 (p. 187)
		Audit Capabilities	AC	J.3 (p. 193)
		Audit Record Structure	ARS	J.4 (p. 199)
		Audit Management	AM	J.5 (p. 203)
Availability (TBD)[a]	—	—	—	K (p. 209)
Group 4				
Security Management	SM	—	—	L (p. 211)
		Secure Installation	SI	L.1 (p. 213)
		Security Policy Selection	SPS	L.2 (p. 217)
		Management of Policy Attributes	MPA	L.3 (p. 223)
		Separation of Administrative Roles	SAR	L.4 (p. 231)
		Security Management Tools	SMT	L.5 (p. 235)

a. Availability requirements were not written for this publication.

At the end of each component subsection is a diagram showing the rating relationships between the example components. Each component of the family is represented as a block in the diagram. Arrows represent the rating relationship between two components. If no arrow connects two components, then no rating relationship exists for those two components (i.e., neither is rated higher). Figure 2 shows the rating relationships for three hypothetical components. The arrow from component **B** to component **A** indicates that **B** is rated higher than **A**.[10] In this figure, component **C** has no rating relationship to either **A** or **B**. In all cases, the relationship between components is derived strictly by the ratings of their constituent elements, which have a well-defined rating relationship.

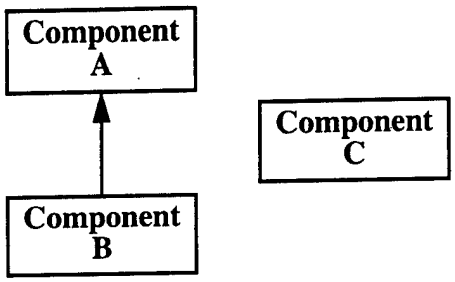


Figure 2. Example Rating Relationships

2.2 ELEMENT AND COMPONENT NAMING

The naming conventions for both the element and component levels are discussed in detail in this section. For element and component naming, abbreviations are used to indicate class and family membership. Table 1 provides the abbreviations for requirements family names, as well as references to the member class for each family. This information should be referenced when reading the element and component sections of this report.

2.2.1 Element Naming

Each requirement element is specified and named independently of the components in which it occurs. For example, our naming convention uses the family abbreviation (e.g., "AP" for Audit Protection) followed by a dash ("-") and terminated by a unique numeric identifier for elements of the same family. Thus, the string **AP-1** may be used to identify such an element.

[10] "Higher" in this sense means that the higher component contains a superset of requirements.

The element naming convention also uses a notion of "variant" forms of element names. Element variants are indicated by a capitalized, single-letter suffix added to the base element name (e.g., AP-1**A**, relating to the previous example). All related elements have the same base identifier (e.g., AP-1). Variant forms indicate an ordering of related elements based on multiple factors (e.g., scope, granularity, strength, and coverage). All elements are uniquely identified with either a base element name or as a variant. It is important to note that while related elements are ordered, there is *no* ordering relationship implied by the base (numeric) identifier values. Only one of these unique element variants (base-ID or variant-ID) is included when a component-level specification is assembled from elements.

The text of the requirements of variant elements is distinguished from the base element in the distributed systems criteria: new (or modified) requirements of the variant appear in boldface, while requirements text that is repeated from the base element appears in a regular font. This convention was used in the presentation of requirements in the Trusted Computer Systems Evaluation Criteria (TCSEC),[11] and its use here is intended to have identical semantics.

Family abbreviations must be unique even across classes. This ensures that elements are uniquely identified. Also, if an element is modified, as it may be when operations such as assignment and refinement are used on an element within a certain component, that modification must be made explicit in referencing the element.

2.2.2 Component Naming

Similar naming conventions are used at the component level. For example, our convention starts with a class abbreviation prefix (e.g., "IA" for Identification and Authentication) followed by an underscore ("_") followed by a family abbreviation (e.g., "UA" for User Authentication) followed by a period (".") and terminated by a unique numeric identifier for components of the same class and family. Thus, the string **IA_UA.1** may be used to identify such a component. Component names can be easily distinguished from element names by the presence of the family abbreviation prefix.

The component naming convention also uses the notion of "variant" forms of component names, similar to that used for elements. For example, the base identifier (i.e., the numeric portion of the component identifier) does *not* indicate leveling. Component vari-

[11] U. S. Department of Defense, National Computer Security Center (NCSC). July 1987. *Trusted Network Interpretation of the Trusted Computer Systems Evaluation Criteria. Version 1.* NCSC-TG-005. Fort George G. Meade, MD: NCSC.

ants are indicated by a capitalized, single-letter suffix added to the base component name (e.g., IA_UA.1**A**, relating to the previous example). Thus, variants of a base component are sequentially ordered from low (the base component) to high (the highest lettered variant). This form is exactly the same as for variants of element names, as described previously; however, the semantics are different for component variant naming.

Variants of a component indicate an ordering, just as for element variant naming. However, for components, this ordering determination is made entirely by examining its composition in terms of elements. One component is a variant of another if it includes equal and higher variants of all the latter's constituent elements. The two related components differ only by variants of the same elements. It follows that if one component contains an element not included in the least variant form within another, the two are not related. Also, for a component to be higher rated, *all* of the elements variants of one of the components must be greater than those of the second component.

Since multiple rating factors (e.g., scope, granularity, strength, and coverage) of the elements included in the components are used to differentiate component ratings, it is highly unlikely that a total order among all possible components could be imposed and reflected in component naming. However, the sequencing of related components (i.e., the base component and its variants) reflects those cases where this ordering can be made explicit. This component rating convention is considered to provide sufficient guidance for component differentiation and selection. While it is often the case that higher-numbered (base) components can be construed to be stronger than lower ones, this is an artifact of the order in which components were constructed and is not intended to be a general property of the naming convention for components.

PART 2. REQUIREMENTS CLASSES

A. TRUSTED COMPUTING BASE CLASS

Families:

1. **Reference Mediation**
2. **Logical TCB Protection**
3. **Physical TCB Protection**
4. **TCB Self-Checking**
5. **TCB Start-Up and Recovery**
6. **TCB Privileged Operation**
7. **TCB Ease-of-Use**

The requirements of *reference mediation* ensure that all references issued by subjects external to a host's TCB (i.e., unprivileged subjects) to other subjects (e.g., to processes, channels), objects, resources, and services of a product are validated by each host's TCB in accordance with the security policies of that host's TCB and the distributed-system product. Satisfying these requirements establishes complete reference mediation (i.e., a reference of a subject external to a TCB cannot circumvent the security policies of that TCB). Functions that implement a security policy provide effective protection against unauthorized access only if all references issued by subjects are directed by TCB code to the appropriate security policy modules for validation. Should such references be incorrectly directed, or not directed at all, to the required policy modules, policy enforcement will be incorrect, incomplete, or absent, despite correct and complete policy implementation. This would allow unprivileged subjects to bypass security policies in a variety of unauthorized ways (e.g., bypass certain access checks for a subset of the objects and subjects, bypass all checks for a type of object whose protection was assumed by applications, retain access rights beyond their intended expiration time, and/or bypass audit). Note that the requirements of the reference mediation are independent of the particular policies supported by a product.

The requirements of *logical TCB protection* ensure that at least one domain is available for a TCB's own execution, and that the TCB is protected from external interference and tampering (e.g., by modification of TCB code or data structures) by unprivileged sub-

jects. The reading and modification of TCB internal variables, that is, variables that are not part of any defined subject or object (e.g., internal TCB buffers, table entries), would not be addressed by low-level product policies defined solely in terms of subjects and objects. In this case, reading of internal TCB variables by users or subjects outside a TCB would not be prohibited, even though it could result in failure to support the organizational policies. Similarly, modification of TCB internal variables may cause (1) the introduction of miscreant code into the TCB, which can modify product policies, (2) the modification of user and application-level objects that depend on the consistency of a TCB's internal variables, (3) denial of service to users and applications, and/or (4) covert transfer of information through a TCB in violation of information-flow policy. Unauthorized acquisition of privileges might allow the reading and modification of TCB internal variables and objects (e.g., password files, group and/or role definition files, files defining security and/or integrity levels) and might allow unprivileged users to execute privileged functions.

To provide TCB isolation, all references to TCB internal entities and all access rights passed by unprivileged subjects to the TCB must be mediated in a non-circumventable manner. This particular form of mediation is not specified as an access mediation requirement because a cyclic dependency would be introduced between access mediation and TCB protection. This is the case because correct reference mediation depends on TCB protection.

Satisfying the requirements of logical TCB protection makes a host TCB self-protecting. Therefore, an unprivileged subject cannot modify or damage a host TCB. The protection of a TCB from external interference and tampering is fundamental to any secure product. Should unprivileged subjects read or modify TCB elements (i.e., data structures and code), the security policy might be circumvented or even modified in potentially undetectable ways. Since physically protected channels are part of TCBs, TCB logical protection requirements also extend to these channels. Logical TCB protection extends to the cryptographic domain whenever this domain is part of the TCB domain. However, additional requirements for the logical protection of the cryptographic domain are included among the requirements of cryptographic support functions whenever the cryptographic domain is a distinct domain of the TCB.

Note that the reference mediation and the logical TCB protection represent the first two requirements of the reference validation mechanism. These two families, as well as the security policy support, are necessary for all protection profiles. The strong dependency of these two families on development assurance is defined by the third requirement of the ref-

erence validation mechanism. See Appendix A for a rationale in the discussion of the Reference Monitor Concept.

The requirements of *physical TCB protection* ensure that the hardware implementing the TCBs and channels is either protected from physical tampering and interference or operated in a protected environment. TCB physical protection requirements refer to restrictions of unauthorized physical access to a TCB and channel hardware, and to deterrence of unauthorized physical use, modification, or substitution of such hardware.

Satisfying the requirements of physical TCB protection causes TCBs and channels to be packaged and used in such a manner that (1) physical tampering is detectable, or (2) resistance to physical tampering is measurable based on defined work factors. Without satisfying these requirements, the protection functions of TCBs and channels lose their effectiveness in environments where physical damage cannot be detected or prevented.

Physical TCB protection extends to the cryptographic domain whenever this domain is part of the TCB domain. However, additional requirements for the physical protection of the cryptographic domain are included among the requirements of the cryptographic support function whenever the cryptographic domain is a distinct domain of the TCB.

The requirements of *TCB self-checking* ensure that hardware, firmware, or software are available to validate the correct operation of TCBs, cryptographic domains, and communication channel hardware and firmware. These requirements also specify validation tests for the consistency of TCBs, cryptographic domain, and channel data structures. TCB, channel, and cryptographic domain self-checking functions are needed to detect the corruption of protection-relevant code and data structures by various failures that do not necessarily stop the product's operation (which would be handled by TCB, channel, and cryptographic-domain recoverability). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes and associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of a TCB, cryptographic domain, or channel due to inadequate physical protection.

Satisfying the TCB self-checking requirements allows the (1) detection of corrupt, protection-relevant code and data structures resulting from various failures, and (2) initiation of corrective action. These requirements are important because corruption of protec-

tion-relevant code and data structures resulting from failures can only be detected, not prevented.

The requirements of *TCB start-up and recovery* ensure that a system is started without protection compromise and can recover without protection compromise after a detected failure or other discontinuity. Start-up and recovery requirements refer to the functions that respond to anticipated failures or discontinuity of operations of TCBs, channels, and cryptographic domains. These requirements cannot include "unanticipated" failures or discontinuity of operation, and manual administrative procedures must be employed for such events. Failures that must be generally anticipated include (1) actions failures (e.g., actions that fail to complete because they detect exceptional conditions during their operation); (2) *unmaskable* action failures that always cause a system crash (e.g., persistent inconsistency of critical system tables, uncontrolled transfers within code caused by transient failures of hardware or firmware, power failures, processor failures); (3) non-volatile media failures causing part or all of the media representing TCB, channel, or cryptographic-domain data to become inaccessible or corrupt (e.g., disk head crash, persistent read/write failure caused by misaligned disk heads, worn-out magnetic coating, dust on the disk surface); and (4) discontinuity of operation caused by erroneous administrative action or lack of timely administrative action (e.g., unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources, inadequate installed configuration). Recovery reconstructs secure states of individual host TCBs, groups of host TCBs, cryptographic domains, and secure channels, or prevent transitions to insecure states, as a direct response to occurrences of expected failures, discontinuity of operation, or start-up. The definition of the secure state for a centralized and distributed-system product is required by these requirements.

Satisfying the requirements of start-up and recovery establishes that the initial and recovered states of individual TCBs, cryptographic domains, and channels, as well as those of the distributed system, satisfy the security policy, reference mediation, and TCB and cryptographic domain protection requirements. These requirements are important because the start-up TCBs, cryptographic domain, and channels' states determine the protection of subsequent states: once the corruption of a protection-relevant data structure by a failure is detected, TCB and channel recovery action becomes necessary.

The requirements of *TCB privileged operation* ensure that host TCB functions operate with the fewest privileges necessary to accomplish their purpose. Functions that limit the privileges available to a host's TCB are primarily intended to limit the damage that can be caused by errors and failures of TCB mechanisms. To accomplish this, it is neces-

sary to limit the interactions among privileged TCB functions to a minimum such that improper use of privileges by a TCB function, module, or action as a consequence of failures or accidents will have limited or no effect on other functions. For example, the association of privileges with different administrative commands facilitates the separation of administrative roles. Similarly, the association of different privileges with separate TCB functions, such as audit trail and password management functions, limits the possibility of unwarranted function interaction. As a consequence, if a penetration of a TCB function takes place, the likelihood that other unrelated functions are also penetrated may be diminished. The finer the granularity of privileges and of privilege association with TCB functions, actions, and administrative roles, the less chance of damage caused by errors, failures, accidents, and penetrations. This is particularly important for security-relevant servers of distributed-system products since the effects of damage in these servers can spread to all host TCBs that depend on these servers.

Satisfying the requirements of TCB privileged operation causes the identification of system privileges required by each TCB function and the addition of mechanisms that associate these privileges with specific TCB functions, modules, or actions. These requirements are important because they help restrict the propagation of errors and failures.

The requirements of *TCB ease-of-use* enable the use of the TCBs and channels of a distributed-system product by users, administrators, and applications. The notion that an information technology (IT) product must include functions which facilitate and enhance the use of basic protection mechanisms is motivated by two related observations. First, if a product's protection mechanisms are complex, difficult to use, or have inadequate performance, they will not be used by system administrators or by application programmers. The mere presence of (potentially elaborate) security policies in a product is insufficient to facilitate the development or use of secure applications and the secure management of a product. An IT product may still be vulnerable to inadvertent errors caused by difficulties in using the product's protection functions. Second, functions that facilitate and enhance the use of basic protection mechanisms may be difficult to retrofit into a product because of their pervasiveness. Instead, to be effective, these requirements must be satisfied in the initial product design.

Satisfying the requirements of TCB ease-of-use provides (1) fail-safe defaults (i.e., defaults that deny access whenever a user or administrator fails to specify access to subjects and objects), (2) user-defined defaults, (3) well-defined interface conventions, (4) the users' capability to reduce their own privileges, and (5) subject, object, resource, and service pro-

tection in common configurations. Without satisfying these requirements, the protection value of the TCB functions is diminished since few users and applications would be able to employ these functions effectively.

1. REFERENCE MEDIATION

1.1 ELEMENTS

RM-1. Specified Reference Mediation

(a) The TCB shall mediate references to the subjects, objects, resources, and services (e.g., TCB functions) described in the TCB specifications.

(b) The mediation shall ensure that all references are directed to the appropriate security-policy functions.

Dependencies:

- Uses: LP-1

RM-2. Reference Mediation to Defined Object Subset

(a) Reference mediation shall include references to the defined subset of subjects, objects, and resources protected under the TCB security policy, and to their policy attributes (e.g., access rights, security and/or integrity levels, role identifiers).

Dependencies:

- Uses: RM-1, LP-1
- Policy: SAO-6

RM-2A. Complete Reference Mediation

(a) Reference mediation shall include **control of** references to **all** subjects, objects, and resources protected under the TCB security policy, and to their policy attributes (e.g., access rights, security and/or integrity levels, role identifiers, quotas).

Dependencies:

- Uses: RM-1, LP-1
- Policy: SAO-6A

RM-2B. Complete Reference Mediation to Object Attributes

(a) Reference mediation shall include control of references to all subjects, objects, and resources protected under the TCB security policy, to their policy (e.g., access

rights, security and/or integrity levels, role identifiers, quotas) **and status** attributes **(e.g., existence, length, locking state)**.

Dependencies:

- Uses: RM-1, LP-1
- Policy: SAO-6B

RM-3. Reference Mediation for Privileged Subjects

(a) References issued by privileged subjects shall be mediated in accordance with the policy attributes defined for those subjects.

Dependencies:

- Uses: RM-1, LP-1
- Policy: ACA-1, ACA-2, ACA-6, PO-1, PO-1A

RM-3A. Model-Based Reference Mediation for Privileged Subjects

(a) References issued by privileged subjects shall be mediated in accordance with the **privilege model** defined for those subjects.

Dependencies:

- Uses: RM-1, LP-1
- Policy: ACA-1, ACA-2, ACA-6, PO-1–PO-3

6.2 COMPONENTS

All components include a common element that specifies the basic semantics of reference mediation. In addition, each component includes two additional elements, each of a distinct type: (1) an element outlining the scope of reference mediation for object accesses, and (2) an element specifying reference mediation for privileged subjects. The five components defined below are rated based on granularity and coverage of the elements in the components.

Component TCB_RM.1. Reference Mediation to Defined Object Subset

This component defines the basic reference mediation requirement, delimits the scope of mediation to a defined object subset, and includes reference mediation for privileged subjects. It consists of the following elements:

- RM-1. Specified Reference Mediation
- RM-2. Reference Mediation to Defined Object Subset
- RM-3. Reference Mediation for Privileged Subjects

Component TCB_RM.1A. Reference Mediation to All Subjects and Objects

This component enhances TCB_RM.1 by extending the scope of reference mediation to include all subjects and objects. It consists of the following elements:

- RM-1. Specified Reference Mediation
- RM-2A. Complete Reference Mediation
- RM-3. Reference Mediation for Privileged Subjects

Component TCB_RM.1B. Reference Mediation to Object Attributes

This component enhances TCB_RM.1A by extending the scope of reference mediation to include not only all subjects and objects but also subject and object status attributes. It consists of the following elements:

- RM-1. Specified Reference Mediation
- RM-2B. Complete Reference Mediation to Object Attributes
- RM-3. Reference Mediation for Privileged Subjects

Component TCB_RM.2. Model-Based Mediation of Privileged Subjects

This component enhances TCB_RM.1A by requiring that the mediation of privileged subject references be governed by a privilege model. It consists of the following elements:

- RM-1. Specified Reference Mediation
- RM-2A. Complete Reference Mediation
- RM-3A. Model-Based Reference Mediation for Privileged Subjects

Component TCB_RM.2A. Model-Based Mediation of All Privileged-Subject References

This component enhances both TCB_RM.1B and TCB_RM.2 in terms of the scope of reference mediation for object accesses and for privileged subject accesses. As such, it represents a superset of both TCB_RM.1B and TCB_RM.2. It consists of the following elements:

- RM-1. Specified Reference Mediation
- RM-2B. Complete Reference Mediation to Object Attributes
- RM-3A. Model-Based Reference Mediation for Privileged Subjects

Component TCB_RM.1 provides a minimal class of reference mediation for profiles whose access control policy covers specified object subsets. Component TCB_RM.1A is useful for profiles where access control policies are mandated for all objects, whereas component TCB_RM.1B is useful for profiles where the access control policy includes information-flow control. Components TCB_RM.2 and TCB_RM.2A are useful for profiles where the TCB is required to satisfy rigorous least-privilege requirements for TCB privileged subjects.

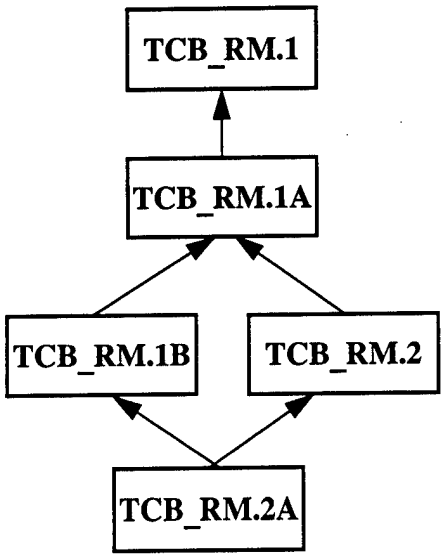


Figure 3. Component Relationships: Reference Mediation

2. LOGICAL TCB PROTECTION

2.1 ELEMENTS

LP-1. TCB Self-Protection

(a) The TCB shall maintain a domain for its own execution that protects it from external interference and tampering (e.g., by reading or modification of its code and data structures).

(b) The protection of the TCB shall provide isolation and noncircumventability as follows:

1. **TCB isolation requires that (1) the address spaces of the TCB and those of unprivileged subjects are separated such that users, or unprivileged subjects operating on their behalf, cannot read or modify TCB data structures or code; (2) the transfers between TCB and non-TCB domains are controlled such that arbitrary entry to, or return from, the TCB is not possible; and (3) the user or application parameters passed to the TCB by addresses are validated with respect to the TCB address space, and those passed by value are validated with respect to the values expected by the TCB.**
2. **Noncircumventability of TCB requires that (1) permissions to objects (and/or to non-TCB data) passed as parameters to the TCB are validated with respect to the permissions required by the TCB, and (2) references to TCB objects implementing TCB isolation functions are mediated by the TCB.**

Dependencies:

- Uses: PP-1

LP-2. Consistency of TCB Global Variables and Operation

(a) TCB protection shall maintain the consistency of TCB global variables and eliminate undesirable dependencies of the TCB operation on unprivileged subject or user actions.

(b) Consistency of TCB global variables requires that consistency conditions defined over TCB internal variables, objects, and functions hold before and after any TCB invocation.

(c) Elimination of undesirable dependencies of the TCB operation on unprivileged subject actions requires that any TCB invocation by an unprivileged subject (or user) input to a TCB call may not place the TCB in a state such that

it is unable to respond to communication initiated by other users.

Dependencies:

- Uses: LP-1, SR-1, SR-2, SR-2A
- Policy: Availability[12]

LP-3. Timing Consistency of TCB Access and Condition Checks

(a) TCB protection shall maintain the timing consistency of access and condition checks.

(b) Timing consistency of access and condition checks requires that a validation check holds at the instant when the TCB action depending on that check is performed.

Dependencies:

- Uses: LP-1

[12] Pending development of availability requirements.

2.2 COMPONENTS

All components include the common element that defines the basic TCB isolation and noncircumventability requirements. The first component consists solely of this common element. Additional components include requirements for mechanisms that ensure the consistency of TCB operation (e.g., global variable and timing consistency). The three logical TCB protection components defined below are rated based on function coverage of the component elements.

Component TCB_LP.1. Protection of the TCB Domain

This component specifies the basic TCB isolation and noncircumventability requirements. It consists of the following element:

- LP-1. TCB Self-Protection

Component TCB_LP.2. TCB Protection with Global Data Consistency

This component extends TCB_LP.1 by including the consistency of TCB global variables and operation. It consists of the following elements:

- LP-1. TCB Self-Protection
- LP-2. Consistency of TCB Global Variables and Operation

Component TCB_LP.3. TCB Protection with Global Data and Timing Consistency

This component extends TCB_LP.2 by including time consistency for security relevant checks. It consists of the following elements:

- LP-1. TCB Self-Protection
- LP-2. Consistency of TCB Global Variables and Operation
- LP-3. Timing Consistency of TCB Access and Condition Checks

Component TCB_LP.1 will be used in the majority of the protection profiles. Components TCB_LP.2 and TCB_LP.3 can be used in profiles where penetration resistance and availability are important.

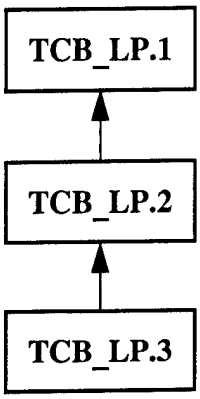


Figure 4. Component Relationships: Logical TCB Protection

3. PHYSICAL TCB PROTECTION

3.1 ELEMENTS

PP-1. Administrative Procedures

(a) Administrative procedures and environmental features necessary for establishing the physical security of a product's TCB shall be defined.

PP-2. Physical TCB Control

(a) Product functions and devices necessary to establish physical control over the product's TCB shall be identified and provided.

Dependencies:

- Uses: PP-1

PP-2A. Tamper-Detection Devices

(a) Product functions and devices necessary to establish physical control over the product's TCB shall be identified and provided.
(b) TCB devices allowing the unambiguous detection of physical tampering shall be employed.
(c) These devices shall be shown to be physically tamper resistant and noncircumventable.

Dependencies:

- Uses: PP-1

PP-2B. Resilient Countermeasures for TCB Protection

(a) Product functions and devices necessary to establish physical control over the product's TCB shall be identified and provided.
(b) TCB devices that provide countermeasures to physical tampering shall be employed.
(c) The strength of these devices shall be determined based on well-defined work factor parameters relevant to the supported policies.
(d) For confidentiality policies, these devices shall resist disclosure via theft, inspection of physical media, wiretapping, and/or analysis of product emana-

tions.

(e) For integrity policies, these devices shall resist modification of hardware functionality and modification of stored data via mechanical methods and/or electronic jamming.

(f) For availability policies, these devices shall resist loss of service via anticipated environmental stress (e.g., water damage, fire, vibration, impact) or other forms of physical attack.

Dependencies:

- Uses: PP-1

3.2 COMPONENTS

All components include the basic requirements for administrative procedures to establish physical TCB security. In addition, all components include requirements for different types of mechanisms that enable physical protection of the TCB. The components defined below are rated based on the coverage and strength of the elements included in the components.

Component TCB_PP.1. Administrative and Environmental Protection

This component includes requirements for administrative procedures to establish physical security and TCB mechanisms to support these procedures. It consists of the following elements:

- PP-1. Administrative Procedures
- PP-2. Physical TCB Control

Component TCB_PP.1A. Detection of Physical Attacks

This component extends TCB_PP.1 by requiring tamper-resistant and noncircumventable devices for detection of physical attacks against the TCB. It consists of the following elements:

- PP-1. Administrative Procedures
- PP-2A. Tamper-Detection Devices

Component TCB_PP.1B. Physical and Environmental Countermeasures

This component extends TCB_PP.1A by requiring resilient countermeasures for physical TCB tampering whose function and strength are dependent on the intended policies supported by the TCB. It consists of the following elements:

- PP-1. Administrative Procedures
- PP-2B. Resilient Countermeasures for TCB Protection

Component TCB_PP.1 will be used in the majority of the protection profiles since most environments require only minimal physical security mechanisms. Component TCB_PP.2 can be used in profiles used for environments where the sensitivity of the applications and data warrant only detection of physical TCB attacks. Component TCB_PP.3 can be used in profiles where physical penetration resistance and availability are important.

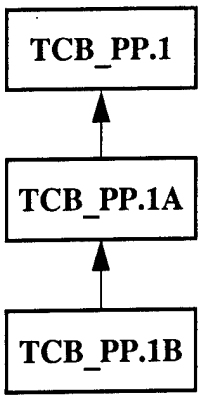


Figure 5. Component Relationships: Physical TCB Protection

4. TCB SELF-CHECKING

4.1 ELEMENTS

SC-1. TCB Hardware Self-Validation

(a) Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Dependencies:

- Uses: PP-1–PP-2B

SC-1A. Test Types for Hardware Self-Validation

(a) Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

(b) These features shall include power-on tests as follows.

1. **The power-on tests shall test all basic components of the TCB hardware and firmware elements including memory boards and memory interconnections, data paths, busses, control logic and processor registers, disk adapters, communication ports, system consoles, and the keyboard speaker.**
2. **These tests shall cover all components that are necessary to run the loadable tests and the operator-controlled tests.**

(c) These features shall also include loadable tests as follows.

1. **The loadable tests shall cover processor components (e.g., arithmetic and logic unit, floating point unit, instruction decode buffers, interrupt controllers, register transfer bus, address translation buffer, cache, and processor-to-memory bus controller), backplane busses, memory controllers, and writable control memory for operator-controlled and remote system-integrity testing.**

(d) These features shall also include operator-controlled tests as follows.

1. **Operator-controlled tests shall be able to initiate a series of one-time or repeated tests, to log the results of these tests, and, if any fault is detected, to direct the integrity-test programs to identify and isolate the failure.**

Dependencies:

- Uses: PP-1–PP-2B

SC-2. TCB Software Self-Validation

(a) Configurable software or firmware features shall be provided that can be used to validate the correct operation of the on-site software elements (i.e., code and data structures) of the TCB.

(b) These features may include, but are not limited to, checksums and consistency checks for TCB elements stored on storage media (e.g., disk-block consistency conditions).

Dependencies:

- Uses: SC-1, SC-1A, LP-1

SC-3. TCB Testing to Detect Transient Failures

(a) Tests that detect possible inconsistencies of the TCB elements (i.e., data structures and code) shall be performed whenever the content or structure of these elements are modified as consequence of a transient failure during an unprivileged subject's action.

Dependencies:

- Uses: SC-1, SC-1A

4.2 COMPONENTS

The components provided for this family include requirements for both TCB hardware and software self-validation, and specify the types of self-validation tests that should be performed. The components defined below are rated based on the coverage of the elements included in the components.

Component TCB_SC.1. Minimal TCB Self-Checking

This component includes a requirement for the basic hardware and software mechanisms to support TCB self-checking. It consists of the following element:

- SC-1. TCB Hardware Self-Validation

Component TCB_SC.1A. Basic TCB Self-Checking

This component extends TCB_SC.1 by specifying a set of test types that are required of TCB self-checking. It consists of the following element:

- SC-1A. Test Types for TCB Hardware Self-Validation.

Component TCB_SC.2. Software-Test Support

This component extends TCB_SC.1A by including requirements for configurable software for TCB self-validation. It consists of the following elements:

- SC-1A. Test Types for TCB Hardware Self-Validation
- SC-2. TCB Software Self-Validation

Component TCB_SC.3. Continuous Software-Test Support

This component extends TCB_SC.2 by requiring that continuous TCB software checks be performed (i.e., whenever TCB elements are modified). It consists of the following elements:

- SC-1A. Test Types for TCB Hardware Self-Validation
- SC-2. TCB Software Self-Validation
- SC-3. TCB Testing to Detect Transient Failures

We anticipate that components TCB_SC.1, TCB_SC.1A, and TCB_SC.2 will be used in the majority of profiles for a wide variety of commercial products (e.g., ranging from personal computers to mainframes, and from local area networks to switching computers of large, geographically distributed networks). Component TCB_SC.3 is intended for use in profiles where availability is important.

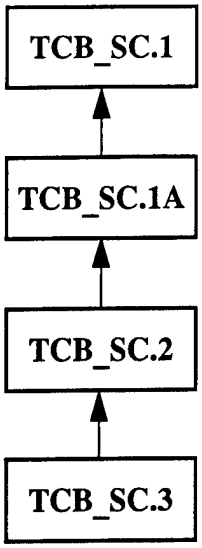


Figure 6. Component Relationships: TCB Self-Checking

5. TCB START-UP AND RECOVERY

5.1 ELEMENTS

SR-1. Secure TCB Recovery

(a) Procedures and/or mechanisms shall be provided to assure that, after a TCB failure or other discontinuity, recovery without protection compromise is obtained.

Dependencies:

- Uses: SC-1–SC-3

SR-2. Administrative Recovery in a Secure State

(a) If automated recovery and start-up is not possible, the TCB shall enter a state where the only system access method is via administrative interfaces, terminals, or procedures.

(b) Administrative procedures shall exist to restore the system to a secure state (i.e., a state in which all the security-policy properties hold).

Dependencies:

- Uses: SC-1–SC-3

SR-2A. Automated Recovery in a Secure State

(a) Automated procedures, under the control of the TCB, shall be provided to assure that after a system failure, other discontinuity, or start-up, a secure state is obtained without undue loss of system or user objects.

(b) The security policy properties, or requirements, used to determine that a secure state is obtained shall be defined.

Dependencies:

- Uses: SC-1–SC-3

SR-2B. Secure State with Object Recovery

(a) Automated procedures, under the control of the TCB, shall be provided to assure that after a system failure, other discontinuity, or start-up, a secure state is obtained

without undue loss of system or user objects.

(b) The security policy properties, or requirements, used to determine that a secure state is obtained shall be defined.

(c) The TCB shall include checkpoint functions for recovery. Upon recovery, it shall be possible to discover which user objects, if any, are corrupted or unaccessible due to the TCB failure and to automatically notify the users.

Dependencies:

- Uses: SC-1–SC-3

SR-2C. Secure State with Object-Loss Minimization

(a) Automated procedures, under the control of the TCB, shall be provided to assure that after a system failure, other discontinuity, or start-up, a secure state is obtained without undue loss of system or user objects.

(b) The security policy properties, or requirements, used to determine that a secure state is obtained shall be defined.

(c) The TCB shall include checkpoint functions for recovery. Upon recovery, it shall be possible to discover which user objects, if any, are corrupted or unaccessible due to the TCB failure and to automatically notify the users.

(d) The TCB functions that can be invoked through the TCB interface shall be atomic (i.e., shall have the property that either their invocation is completed correctly or the recovered system state should be the one immediately prior to the execution of the TCB function).

(e) The recovered secure state should minimize the corruption and inaccessibility of user objects due to the TCB failure.

Dependencies:

- Uses: SC-1–SC-3

5.2 COMPONENTS

All components include a basic requirement that defines in a generic way the notion of secure (or trusted) TCB recovery. The components provided below suggest that four types of secure recovery should be identified beyond the secure recovery element included in the first component. The four non-minimal components deal with recovery in a secure state (i.e., a state in which all security policy properties hold) (1) via administrative means, (2) via automated means, (3) with object loss notification, and (4) with object loss minimization. The components defined below are rated based on the coverage and strength of the elements included in the components.

Component TCB_SR.1. Minimal Start-up and Recovery Functions

This component includes a requirement for the basic requirement to support secure (trusted) TCB start-up and recovery. It consists of the following element:

- SR-1. Secure TCB Recovery

Component TCB_SR.2. Basic Start-up and Recovery Functions

This component extends TCB_SR.1 by including a requirement for TCB start-up and recovery in a secure state via administrative means. It consists of the following elements:

- SR-1. Secure TCB Recovery
- SR-2. Administrative Recovery in a Secure State

Component TCB_SR.2A. Automated Start-up and Recovery Functions

This component extends TCB_SR.2 by including a requirement for TCB start-up and recovery in a secure state using automated procedures. It consists of the following elements:

- SR-1. Secure TCB Recovery
- SR-2A. Automated Recovery in a Secure State

Component TCB_SR.2B. Start-up and Recovery with Object-Loss Detection

This component extends TCB_SR.2A by including a requirement for TCB start-up and recovery in a secure state with object loss detection and notification. It consists of the following elements:

- SR-1. Secure TCB Recovery

- SR-2B. Secure State with Object Recovery.

Component TCB_SR.2C. Start-up and Recovery with Object-Loss Minimization

This component extends TCB_SR.2B by including a requirement for TCB start-up and recovery in a secure state with object loss minimization. It consists of the following elements:

- SR-1. Secure TCB Recovery
- SR-2C. Secure State with Object-Loss Minimization

We anticipate that components TCB_SR.1 and TCB_SR.2 will be used in the majority of profiles for typical small configurations where scalable, timely recovery is not required (e.g., availability policies are not considered necessary). In contrast, component TCB_SR.2A can be used in profiles for environments where system size and response time constraints rule out manual recovery of a secure state. Component TCB_SR.2B can be used in profiles for environments where the detection and user notification of object losses during secure (trusted) recovery are required. This component helps ensure that trivial forms of secure states (e.g., via object destruction) become known to users and administrators, thereby giving them a chance to use manual recovery procedures. Component TCB_SR.2C can be used in profiles for environments where object losses during secure (trusted) recovery must be minimized (i.e., environments where availability policies are considered to be important to system operation).

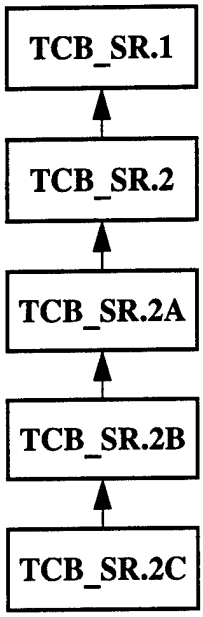


Figure 7. Component Relationships: TCB Start-Up and Recovery

6. TCB PRIVILEGED OPERATION

6.1 ELEMENTS

PO-1. Identification and Definition of TCB Privileges

(a) TCB privileges needed by individual functions, or groups of functions, shall be identified.

(b) Privileged TCB calls or access to privileged TCB objects, such as user and group registration files, password files, security and integrity-level definition files, role definition files, or audit-log files, shall also be identified.

Dependencies:

- Uses: LP-1
- Policy: ACA-1, ACA-2, ACA-6

PO-1A. Definition of TCB-Function and Administrative Privileges

(a) TCB privileges needed by individual functions, or groups of functions, shall be identified.

(b) Privileged TCB calls or access to privileged TCB objects, such as user and group registration files, password files, security and integrity-level definition files, role definition files, or audit-log files, shall also be identified.

(c) It shall be possible to associate TCB privileges with TCB operations performed by administrative users.

Dependencies:

- Uses: PO-1, LP-2
- Policy: ACA-1, ACA-2, ACA-6

PO-2. Least Privilege for TCB Functions

(a) The identified privileged functions of a TCB shall be associated only with the privileges necessary to complete their task.

Dependencies:

- Uses: PO-1, LP-2

PO-2A. Least Privilege for Modules of TCB Functions

(a) The modules of a TCB function shall be associated only with the privileges necessary to complete their task.

Dependencies:

- Uses: PO-1, LP-2

PO-2B. Least Privilege for Actions of TCB Functions

(a) The modules of a TCB function shall be associated only with the privileges necessary to complete their task.

(b) TCB privileges needed by individual actions of a module (i.e., function invocations) shall be identified (e.g., privileges shall be assigned to actions that bypass access controls, such as disclosure and modification of user objects).

(c) Each action shall be associated only with the privileges necessary to complete its task.

Dependencies:

- Uses: PO-1, LP-2

PO-3. Mandated Use of TCB Privileges

(a) The identified TCB privileges shall be used by each function, module, or action to restrict the propagation of errors and failures of security mechanisms that may lead to protection policy violations. In particular:

1. **TCB mechanisms allowing each function, module, or action to acquire individual privileges up to the maximum necessary and allowed, and to drop those privileges (e.g., functions implementing privilege bracketing) shall be defined; and**
2. **These mechanisms shall be used to limit the use of privileges that allow the bypassing of security policy controls within the TCB.**

Dependencies:

- Uses: PO-1, LP-2

PO-4. Implementation Support for Module Privileges

(a) Support for system privilege implementation and association with TCB modules provided by lower-level mechanisms or procedures (e.g., operating system, processors, language) shall be provided.

Dependencies:

- Uses: PO-1, LP-2

PO-4A. Implementation Support for Action Privileges

(a) Support for product privilege implementation and association with TCB **actions** provided by lower-level mechanisms or procedures (e.g., operating system, processors, language) shall be provided.

Dependencies:

- Uses: PO-1, LP-2

6.2 COMPONENTS

The components of this family include three types of requirements, namely (1) identification of privileges that authorize TCB operations, (2) granularity of the association between privileges and TCB entities (e.g., functions, modules, actions, privileged objects), and (3) implementation support for TCB privileges. The components provided below illustrate some of the uses of the above requirement types. The components defined below are rated based on the granularity and coverage of the individual elements.

Component TCB_PO.1. Privilege Association with TCB Functions

This component defines the basic requirement to identify and associate privileges with TCB functions. It consists of the following elements:

- PO-1. Identification and Definition of TCB Privileges
- PO-2. Least Privilege for TCB Functions

Component TCB_PO.2. Privilege Association with TCB Modules

This component extends TCB_PO.1 by requiring that (1) TCB privileges be associated with TCB operations performed by administrative users, not just with TCB functions; (2) privileges be associated with TCB modules (i.e., with a lower granularity entity of the TCB); and (3) support for the privilege-module association be provided by lower-level mechanisms of the TCB. This component consists of the following elements:

- PO-1A. Definition of TCB-Function and Administrative Privileges
- PO-2A. Least Privilege for Modules of TCB Functions
- PO-4. Implementation Support for Module Privileges

Component TCB_PO.2A. Privilege Association with TCB Actions

This component extends TCB_PO.2 by requiring that (1) privileges be associated with TCB actions (i.e., with a lower granularity entity of the TCB), and (2) support for the privilege-action association be provided by lower-level mechanisms of the TCB. This component consists of the following elements:

- PO-1A. Definition of TCB-Function and Administrative Privileges
- PO-2B. Least Privilege for Actions of TCB Functions
- PO-4A. Implementation Support for Action Privileges

Component TCB_PO.3. Dynamic Privilege Association with Individual TCB Actions

This component extends TCB_PO.2A by requiring that defined privileges be associated with TCB actions in a manner that would restrict propagation of errors and failures of security mechanisms within the TCB. This component consists of the following elements:

- PO-1A. Definition of TCB-Function and Administrative Privileges
- PO-2B. Least Privilege for Actions of TCB Functions
- PO-3. Mandated Use of TCB Privileges
- PO-4A. Implementation Support for Action Privileges

We anticipate that the above components will be used in profiles for environments where a significant degree of resistance against penetration and failure effects is required. We envision that component TCB_PO.1 will be used in most commercial products where different administrative roles are indistinguishable. In contrast, component TCB_PO.2 can be used in profiles for environments where it is important to separate administrative privileges and duties and to provide a fine granularity of privilege association with TCB entities. Components TCB_PO.2A and TCB_PO.3 can be used in profiles for environments where explicit TCB mechanisms are needed for limiting the propagation of errors and failure effects and where a very fine granularity of privileges (i.e., least privileges) should be associated with TCB entities. These components could be used in environments where the high integrity of TCB operation is deemed important.

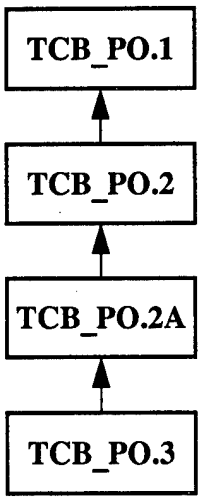


Figure 8. Component Relationships: TCB Privileged Operation

7. TCB EASE-OF-USE

7.1 ELEMENTS

EU-1. Administrative Support

(a) The TCB shall provide well-defined actions to undertake administrative functions.

(b) Default options shall be provided for security parameters of administrative functions.

Dependencies:

- Uses: LP-1

EU-1A. Administrative Support with Fail-Safe Defaults

(a) The TCB shall provide well-defined actions to undertake administrative functions.

(b) Default options shall be provided for security parameters of administrative functions.

(c) The TCB shall include fail-safe defaults for the policy attributes of the defined subjects and objects, as well as user-settable defaults for the defined subjects and objects.

Dependencies:

- Uses: LP-1

EU-1B. Administrative Support with Complete Fail-Safe Defaults

(a) The TCB shall provide well-defined actions to undertake administrative functions.

(b) **Fail-safe** default options shall be provided for security parameters of administrative functions.

(c) The TCB shall include fail-safe defaults for the policy attributes of **all** subjects, objects **(e.g., devices), and services used in common system configurations,** as well as user-settable defaults for **these** subjects and objects.

Dependencies:

- Uses: LP-1

EU-1C. Administrative Support with User-Settable, Fail-Safe Defaults

(a) The TCB shall provide well-defined actions to undertake administrative functions.
(b) Fail-safe default options shall be provided for security parameters of administrative functions.
(c) The TCB shall include fail-safe, user-settable defaults for the policy attributes of all subjects, objects (e.g., devices), and services.

Dependencies:

- Uses: LP-1

EU-2. Applications Support

(a) The TCB shall provide well-defined programming interfaces and programming functions (e.g., libraries) for all its policies to support the development of applications that can define and enforce security policies on application-controlled subjects and objects.
(b) The TCB shall enable user-controlled reduction of access rights available to applications.

Dependencies:

- Uses: EU-1–EU-1C

7.2 COMPONENTS

The components of this family include two types of elements, namely (1) definition of administrative functions and their default parameters, and (2) definition of programming interfaces and functions for development of secure applications. Different types of defaults for administrative functions provide the main rating factor for these components. These components are rated based on the coverage of the individual elements.

Component TCB_EU.1. Ease of Security Administration

This component includes the basic requirement for security administration. It consists of the following element:

- EU-1. Administrative Support

Component TCB_EU.2. Ease of Application Programming

This component extends TCB_EU.1 by including a requirement for the use of fail-safe defaults for a defined set of subjects and objects, and a requirement for application support. It consists of the following elements:

- EU-1A. Administrative Support with Fail-Safe Defaults
- EU-2. Applications Support

Component TCB_EU.2A. Ease of Security Use in Common Configurations

This component extends TCB_EU.2 by extending the requirement for the use of fail-safe defaults to all subjects and objects in common configurations. It consists of the following elements:

- EU-1B. Administrative Support with Complete Fail-Safe Defaults
- EU-2. Applications Support

Component TCB_EU.2B. Ease of Security Use in All Configurations

This component extends TCB_EU.2A by the requirement for user-settable, fail-safe defaults to all subjects and objects. It consists of the following elements:

- EU-1C. Administrative Support with User-Settable, Fail-Safe Defaults
- EU-2. Applications Support

We anticipate that the first two components, TCB_EU.1 and TCB_EU.2, will be used in most profiles for low-end security systems and products. The first component can be used for turnkey systems where no application development or integration needs to be performed, whereas the second component can be used where support for secure applica-

tion development becomes necessary. Component TCB_EU.2A can be used in profiles for environments where the access controls need to be applied to all subjects and objects in common system configurations (i.e., in systems where the use of security features is not an exception). The last component can be used for high-end security systems where default setting can be controlled by users on an application basis.

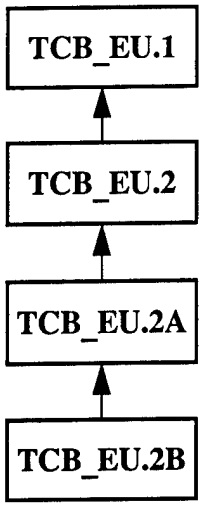


Figure 9. Component Relationships: TCB Ease-of-Use

B. IDENTIFICATION AND AUTHENTICATION CLASS

Families:

1. **Identification**
2. **Channel Authentication**
3. **User Authentication**
4. **Inter-Realm Authentication**
5. **Authentication Policy**

Identification assigns a unique, unambiguous name or identifier to all subjects (e.g., users, communication channels) that perform any action that must be mediated within the system. Authentication functions attribute responsibility for an action to an identified subject. These actions are typically invoked by requests on a channel. For example, user authentication involves the verification of the user identity claimed during a login request on a login channel. For other types of channels, authentication simply determines the identity of the subjects at one or both ends of a channel. Identification and authentication (I&A) requirements refer to complete and unambiguous subject identification, to user and channel authentication, to inter-realm authentication, and to authentication policy. The distinction between users and channels for subject I&A is fundamental because, unlike other subjects, users cannot typically remember large, secret numbers and characters, cannot perform their encryption, and cannot respond in real-time to authentication challenges issued via a channel.

Channel authentication specifies functional requirements for desirable authentication properties that counter common threats. For example, basic authentication helps establish (1) the freshness of authentication messages in face of message replay and re-ordering attacks, and (2) the presence of the authenticated subject at the end of a channel, in addition to establishing the identity of the original creator of a channel message. Limited-time authentication helps limit the damage of compromised authentication, whereas non-repudiation helps obtain evidence confirming the origin of a message to a third party, which may become necessary in case of authentication disputes. Other requirements, such as com-

pound subject authentication and anonymous (but traceable) authentication, extend the authentication function to different applications.

User authentication specifies functional requirements to verify the claimed identity of individuals attempting system entry. Identification and authentication are required to ensure that the authenticated users are associated with the proper set of policy attributes (e.g., identity, groups, roles, security or integrity levels, time intervals, location). Thus, identification and authentication establish that all individuals entering a system and accessing its subjects, objects, and services are authorized to do so by the system entry and protection policy, and that the accountability policy can be enforced. In operating systems, the user I&A functions constitute the main part of the process commonly known as "login," with the balance of the process consisting of system entry and trusted path functions.

Inter-realm authentication refers to the avoidance of globally trusted authentication authorities. *Authentication policy* captures requirements specifying desired authentication objectives.

1. IDENTIFICATION

1.1 ELEMENTS

IAI-1. Identification of Simple Subjects

(a) All types of simple subjects that must be authenticated shall be identified. A simple subject can be a process, a group of users, a machine, a communication channel, a realm, a service, or a program.

(b) The distributed system shall have the capability to authenticate simple subjects as required by the system security policy.

Dependencies:

- Uses: LP-1

IAI-1A. Identification of Compound Subjects

(a) All types of simple subjects that must be authenticated shall be identified. A simple subject can be a process, a group of users, a machine, a communication channel, a realm, a service, or a program.

(b) The distributed system shall have the capability to authenticate simple subjects as required by the system security policy.

(c) The distributed system shall also be capable of supporting compound subjects. These compound subjects include delegation chains, restricted delegation chains, and conjunctions of subjects (i.e., AND-chained identities), as required by the system security policy.

Dependencies:

- Uses: LP-1

IAI-2. Complete and Unambiguous Identification

(a) The identification of each user and subject must be complete (i.e., all users and subjects, including privileged ones, must be identified).

(b) The identification of each user and subject must be unambiguous (i.e., every user and every subject must have an identity that is different from that of any

other user or subject, and this identity shall not be reused).

Dependencies:

- Uses: IAI-1, IAI-1A

IAI-3. Mandated Identification and Authentication

(a) All users shall be required to identify and authenticate themselves before beginning to perform any other actions that must be mediated.

Dependencies:

- Uses: IAI-1, IAI-1A

IAI-4. Auditability of User Actions

(a) User and subject identification shall provide the capability of associating the unique user identity with all auditable actions taken by an individual.

Dependencies:

- Uses: IAI-1, IAI-1A

3.2 COMPONENTS

All possible components using the identification functions listed above must include (1) identification of simple or compound objects, (2) complete and unambiguous identification, and (3) mandated I&A elements. Whether the last element is also necessary depends on whether audit is a required feature of the protection profile under construction. The identification components are rated based on the coverage of the individual element of the identification function and the scope of subjects.

Component IA_IAI.1. Identification of Simple Subjects

This component includes all the basic requirements for the identification of simple subjects. It consists of the following elements:

- IAI-1. Identification of Simple Subjects
- IAI-2. Complete and Unambiguous Identification
- IAI-3. Mandated Identification and Authentication

Component IA_IAI.1A. Identification of Compound Subjects

This component includes all the basic requirements for the identification of both simple and compound subjects. It consists of the following elements:

- IAI-1A. Identification of Compound Subjects
- IAI-2. Complete and Unambiguous Identification
- IAI-3. Mandated Identification and Authentication

Component IA_IAI.2. Identification of Simple Subjects

This component extends IA_IAI.1 with the requirement of auditability of identified-user actions. It consists of the following elements:

- IAI-1. Identification of Simple Subjects
- IAI-2. Complete and Unambiguous Identification
- IAI-3. Mandated Identification and Authentication
- IAI-4. Auditability of User Actions

Component IA_IAI.2A. Identification of Compound Subjects

This component extends IA_IAI.2 with the requirement of auditability of identified-user actions. It consists of the following elements:

- IAI-1A. Identification of Compound Subjects
- IAI-2. Complete and Unambiguous Identification

- IAI-3. Mandated Identification and Authentication
- IAI-4. Auditability of User Actions

It is envisioned that components IA_IAI.1 and IA_IAI.2 will be used in most profiles for environments where only simple subjects are needed. IA_IAI.2 can be used where auditability of user actions is necessary, whereas IA_IAI.1 can be used in profiles of special systems and products where subject auditability is unessential. Components IA_IAI.1A and IA_IAI.2A parallel the elements of IA_IAI.1 and IA_IAI.2 for profiles where support for compound subjects is required.

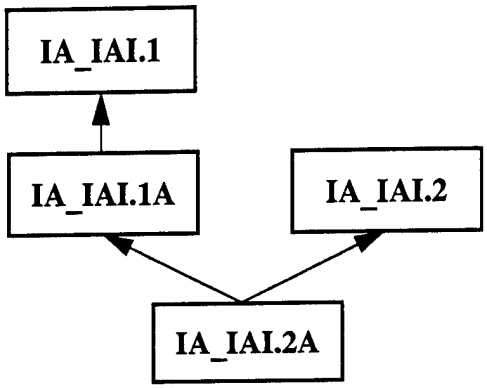


Figure 10. Component Relationships: Identification

2. CHANNEL AUTHENTICATION

2.1 ELEMENTS

CA-1. Channel Property

(a) Channels shall have the property that whenever a subject receives a message, the subject can identify the channel on which the message arrives and the direction of the message.

Dependencies:

- Uses: IAI-1, SCF-2–SCF-5, CDP-2, CDP-3, SKM-1–SKM-11

CA-2. Message-Origin Authentication

(a) The distributed system shall be able to perform message-origin authentication on a channel (i.e., whenever a subject receives a message, it can know which subject originally created that message).

(b) The message-origin authentication mechanism shall establish that a channel message is not a replay of other messages originated earlier.

Dependencies:

- Uses: CA-1, SCF-2–SCF-5, CDP-2, CDP-3, SKM-1–SKM-11

CA-3. Support for Channel Authentication

(a) The distributed system shall have the capability to provide authentication channels for all communications.

Dependencies:

- Uses: CA-1

CA-4. Mutual Authentication

(a) The distributed system shall be able to perform mutual authentication on a channel (i.e., whenever two subjects exchange messages with each other, each recipient can know that the received message was originally created by the other subject as part of that message exchange).

(b) The mutual authentication mechanism shall establish that a channel message

is not a replay of other messages originated earlier.

Dependencies:

- Uses: CA-1, SCF-2–SCF-5, CDP-2, CDP-3, SKM-1–SKM-11

CA-5. Revocation of Channel Authentication

(a) Channel authentication must be revocable (e.g., administrators must be able to revoke authentication).

(b) If revocation is not immediate, a capability to specify by when revocation takes effect shall be provided (e.g., immediate revocation need not be required).

Dependencies:

- Uses: CA-1
- Policy: SKM-10, SKM-10A, SKM-10B

CA-6. Finite Authentication Duration

(a) The authentication of each channel shall be valid only for a specified time duration unless explicit renewal or extension action is taken (e.g., duration is limited by the channel-key lifetime and its renewal).

Dependencies:

- Uses: CA-1
- Policy: SKM-10, SKM-10A, SKM-10B

CA-7. Validity of Authentication Data

(a) If data are supplied by a third party as part of the authentication process, these data shall have an interval of validity specified (e.g., postdated authentication).

(b) An upper bound on the interval of validity shall be administratively specified and enforced.

Dependencies:

- Uses: CA-1
- Policy: SKM-10, SKM-10A, SKM-10B

CA-8. Delegation Chain Support

(a) A subject shall be able to delegate a subset of its policy attributes, based on a security policy, to another subject; the latter subject can further delegate its policy attributes, or a subset thereof, together with the received attributes, or a subset thereof, to another subject; and so on, thereby forming a delegation

chain.

(b) Delegation chains shall be able to preserve the distinction between the identity of the original subject and that of delegates.

Dependencies:

- Uses: CA-1

CA-9. Delegation Chain Authentication

(a) Whenever a subject authenticates a delegation chain at an end of a channel, that subject shall be able to derive authentication evidence confirming that each delegator subject in the chain delegated a subset of its policy attributes to its successor.

(b) Whenever a subject authenticates a delegation chain at an end of a channel, that subject shall be able to derive authentication evidence confirming that each subject being delegated to explicitly accepted delegation.

Dependencies:

- Uses: CA-1, CA-8, SCF-2–SCF-5, CDP-2, CDP-3, SKM-1–SKM-11

CA-10. Restricted Delegation Chains

(a) The subject initiating the delegation chain shall be able to restrict the validity of the policy-attribute delegation to a specific service, object, or object permission (i.e., the subject shall be able to create a restricted delegation chain).

Dependencies:

- Uses: CA-8

CA-11. Authenticating AND-Chains on a Channel

(a) Whenever a subject authenticates an AND-chained set of identities at the end of a channel, that subject shall be able to authenticate each identity of the chain individually.

Dependencies:

- Uses: CA-9, SCF-2–SCF-5, CDP-2, CDP-3, SKM-1–SKM-11

CA-12. Non-Repudiation Support

(a) Functions shall be provided to protect a subject that participates in a communication via a given channel, or set of channels, from false denial of participation in that communication by another subject.

(b) All subjects that participate in a communication shall (1) be identified, and (2) have a unique, protected, and auditable mapping between their identifiers and

legal names, which is known to all participants.

(c) Authorized subjects shall be able to define, associate, maintain, and revoke:

1. **The authority (context) and signature of a communication originator; and**
2. **The authority and signature of dispute-resolution subjects that are trusted by all communication participants.**

(d) Recipients of a communication shall be able to interpret the authority (context) and signature of a communication originator.

(e) Functions shall be provided to enable an originator or a recipient of a communication message to establish (1) that the message semantics are consistent with the authority and signature of the communication originator, and (2) that the message was in its possession at a fixed point in time.

(f) These functions shall be able to associate with the message a time (and date) stamp provided by a source considered trustworthy by the originator, recipient, and dispute-resolution subjects.

Dependencies:

- Uses: IAI-1–IAI-3, AE-1–AE-2, DIF-4–DIF-6A, CA-1–CA-3, CA-5–CA-7

CA-13. Display of Non-Repudiation Data

(a) Functions shall be available that provide a channel to display a message, and all non-repudiation data including the following:

1. **The identity of the originator and, if required, the identity of recipients, as well as the data necessary for authenticating these identities; e.g., when making use of public-key certificates, functions must be provided to display the authentication path needed to validate these identities and the associated certification revocation status information;**
2. **Any per-originator authority (context) and signature data, revocation time data, explicit message semantics, and explicit dispute-resolution authority and signature data; and**
3. **Message timestamps, if multiple timestamps are associated with a message, and the identity of the timestamp authority.**

(b) The display functions shall be available to the message originator, recipients, and dispute-resolution subjects.

(c) If a system provides trusted channels, these channels shall be used for display of non-repudiation messages and data.

Dependencies:

- Uses: DTP-1–DTP-3, CA-12

CA-14. Accountable Use of Non-Repudiation Functions.

(a) Functions shall be provided to enable a communication originator to exercise

positive control over the invocation of non-repudiation facilities; e.g., prior to affixing a digital signature to a message for non-repudiation purposes, the originator may be required to acknowledge this action explicitly.

(b) A trusted channel shall be used for the invocation of non-repudiation services by users, if trusted channels are available.

(c) The invocation of a non-repudiation service by a message originator shall be auditable.

Dependencies:

- Uses: DTP-1–DTP-3, CA-12

2.2 COMPONENTS

The components presented below partition the I&A elements into four classes, namely (1) basic elements that are common to all components; (2) extended elements, which include non-repudiation and traceable authentication; (3) delegation support elements; and (4) joint (AND-chained) identity authentication. Other ways of partitioning the I&A elements are undoubtedly possible; lower-granularity components are also possible. The rating of the components presented below is based on the coverage of the channel authentication elements and scope of subjects.

Component IA_CA.1. Basic Channel Authentication

This component consists of the basic elements found in most authentication subsystems of distributed systems and networks. It includes both the main authentication requirements (e.g., message-origin and mutual authentication) and their features (e.g., revocable authentication, finite-duration authentication, and validated authentication options and parameters). This component consists of the following elements:

- CA-1. Channel Property
- CA-2. Message-Origin Authentication
- CA-3. Support for Channel Authentication
- CA-4. Mutual Authentication
- CA-5. Revocation of Channel Authentication
- CA-6. Finite Authentication Duration
- CA-7. Validity of Authentication Data

Component IA_CA.2. Authentication with Delegation

This component extends IA_CA.1 by including delegation requirements. It consists of the following elements:

- CA-1. Channel Property
- CA-2. Message-Origin Authentication
- CA-3. Support for Channel Authentication
- CA-4. Mutual Authentication
- CA-5. Revocation of Channel Authentication
- CA-6. Finite Authentication Duration
- CA-7. Validity of Authentication Data
- CA-8. Delegation Chain Support

- CA-9. Delegation Chain Authentication
- CA-10. Restricted Delegation Chains

Component IA_CA.3. Authentication with Delegation and Joint Identities

This component extends IA_CA.2 by including requirements for joint-identity (AND-chain) authentication. It consists of the following elements:

- CA-1. Channel Property
- CA-2. Message-Origin Authentication
- CA-3. Support for Channel Authentication
- CA-4. Mutual Authentication
- CA-5. Revocation of Channel Authentication
- CA-6. Finite Authentication Duration
- CA-7. Validity of Authentication Data
- CA-8. Delegation Chain Support
- CA-9. Delegation Chain Authentication
- CA-10. Restricted Delegation Chains
- CA-11. Authenticating AND-Chains on a Channel

Component IA_CA.4. Extended Authentication

This component extends IA_CA.3 by including requirements for non-repudiable authentication. This component consists of all the channel-authentication elements:

- CA-1. Channel Property
- CA-2. Message-Origin Authentication
- CA-3. Support for Channel Authentication
- CA-4. Mutual Authentication
- CA-5. Revocation of Channel Authentication
- CA-6. Finite Authentication Duration
- CA-7. Validity of Authentication Data
- CA-8. Delegation Chain Support
- CA-9. Delegation Chain Authentication
- CA-10. Restricted Delegation Chains
- CA-11. Authenticating AND-Chains on a Channel
- CA-12. Non-Repudiation Support

- CA-13. Display of Non-Repudiation Data
- CA-14. Accountable Use of Non-Repudiation Functions

It is envisioned that component IA_CA.1 will be used in profiles requiring authentication of simple subjects and basic functions such as revocable and finite-time authentication. Component IA_CA.2 can be used in profiles requiring support for delegation chains, whereas component IA_CA.3 can be used in profiles requiring authentication for each subject in the delegation chain. Component IA_CA.4 can be used wherever non-repudiated authentication is required.

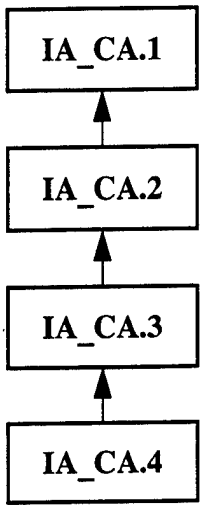


Figure 11. Component Relationships: Channel Authentication

3. USER AUTHENTICATION

3.1 ELEMENTS

UA-1. User-Authentication Support

(a) User-authentication functions shall be provided by the TCBs of centralized-system products.

(b) In distributed systems, these functions shall be provided by a set of TCBs interconnected by secure channels.

Dependencies:

- Uses: LP-1, PP-2–PP-2B, IAI-1

UA-2. User-Authentication Data

(a) User-authentication functions shall maintain authentication data that includes information for verifying the identity of individual users (e.g., passwords, keys, key seed data).

(b) These data shall be used to authenticate the user's identity.

Dependencies:

- Uses: UA-1, SKM-1–SKM-4

UA-3. Protection of User-Authentication Data

(a) User-authentication functions shall protect authentication data to prevent a user from masquerading as another user.

Dependencies:

- Uses: UA-1, SKM-8–SKM-11, LP-1, PP-2–PP-2B

UA-4. Handling of Repeated Authentication Failures

(a) The user-authentication functions of an individual TCB shall end the attempted login session if the user performs the authentication procedure incorrectly for a number of successive times (i.e., a threshold) specified by an authorized

administrator.

(b) A default threshold shall be defined.

(c) When the threshold is exceeded, the authentication function shall send an alarm message to an authorized administrator (e.g., to the administrator's mailbox, workstation, console), log this event in the audit trail, and delay the next login by an interval of time specified by an authorized administrator.

(d) A default time interval shall be defined.

(e) User-authentication functions of that TCB shall provide the option to disable the user identity or account when the threshold of successive, unsuccessful log-in attempts is violated more than a number of times specified by an authorized administrator.

(f) The option to disable the user identity or account shall be disabled by default, as it may cause unauthorized denial of service.

Dependencies:

- Uses: UA-1

UA-5. Minimizing Exposure of Authentication Data

(a) The authentication functions shall minimize the exposure of user-authentication data (e.g., passwords, secret or private keys) to decrease the possibility of unauthorized disclosure, modification, deletion, substitution, or use.

Dependencies:

- Uses: UA-1, SKM-8–SKM-11, LP-1, PP-2–PP-2B

UA-6. Minimizing Authentication Data

(a) The number of secret or private authentication data copies shall be minimized subject to availability constraints.

(b) If sharing of authentication data among TCBs is required, it shall be minimized by identifying specific trusted authentication authorities (i.e., trusted third parties, which are separate TCBs in their realms) with which other TCBs can share these data on a pairwise-private basis.

(c) The trusted third-party TCBs shall be protected from external interference and tampering.

Dependencies:

- Uses: UA-1, SKM-8–SKM-11, LP-1, PP-2–PP-2B

UA-7. Maintenance of Authenticated User Status

(a) User-authentication functions of a TCB shall have the capability to maintain, protect, and display status information for all active users of that TCB (e.g.,

users currently logged on, current policy attributes) and of all user accounts (i.e., enabled or disabled user identity or account).

(b) It shall be possible to limit access to authenticated user status to authorized administrators.

Dependencies:

- Uses: UA-1, LP-1, PP-2–PP-2B

UA-8. Support for Multiple User-Authentication Mechanisms

(a) User-authentication functions shall be able to support multiple authentication mechanisms, such as token-based cards, smart-cards, or trusted third-party mechanisms, in the place of or in addition to the default authentication (e.g., password-based) mechanism, to authenticate the user.

(b) These functions shall be able to enforce separate user authentication procedures based on specific policy attributes (e.g., login via remotely located systems shall require token-based cards; or login with certain groups, roles, and security levels shall require smart cards).

Dependencies:

- Uses: UA-1

UA-9. Multiple User Authentication

(a) It shall be possible to authenticate each user by two or more types of authentication mechanisms; i.e., the authentication is successful only if all mechanisms individually indicate successful authentication.

(b) User-authentication functions shall be able to enforce the use of these mechanisms on a policy-attribute basis.

Dependencies:

- Uses: UA-1

UA-10. Single-User Login

(a) User-authentication functions shall be able to support single-user login regardless of the number of realms or per-realm hosts in the distributed system.

Dependencies:

- Uses: UA-1

3.2 COMPONENTS

All user authentication components include a set of three basic elements, namely (1) user authentication support, which sets the basic user authentication requirement; (2) user authentication data, which requires per-user maintenance of authentication data; and (3) protection of the user authentication data, which motivates the requirement for authentication data protection. Additional components include elements of exception handling, minimization of authentication data exposures, and authentication data copies to provide additional measures of data protection. Additional strength in user authentication can be gained by the use of multiple mechanisms and multiple user-authentication procedures, as required by some of the other components defined below. Finally, single login functions are suggested for distributed systems both as an ease-of-use function and as a means to limit the number of times a user or a user application must login to start a computation on a remote host. The user authentication components below are rated based on the coverage and strength of the included elements.

Component IA_UA.1. Minimal User Authentication

This component includes the minimal requirements to support basic user authentication. It consists of the following elements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data

Component IA_UA.2. Basic User Authentication

This component extends IA_UA.1 by including requirements for authentication exception response to guard against guessing authentication data, and maintenance and protection of authenticated user status. It consists of the following elements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data
- UA-4. Handling of Repeated Authentication Failures
- UA-7. Maintenance of Authenticated User Status

Component IA_UA.3. Extended User Authentication

This component extends IA_UA.2 by including additional requirements for protecting authentication data. It consists of the following requirements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data
- UA-4. Handling of Repeated Authentication Failures
- UA-5. Minimizing Exposure of Authentication Data
- UA-6. Minimizing Authentication Data
- UA-7. Maintenance of Authenticated User Status

Component IA_UA.4. Multiple User-Authentication Mechanisms

This component strengthens the user-authentication features of IA_UA.3 by including a requirement for supporting multiple authentication mechanisms based on specific policy attributes, such as location of remote login, security level, and roles. This component consists of the following elements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data
- UA-4. Handling of Repeated Authentication Failures
- UA-5. Minimizing Exposure of Authentication Data
- UA-6. Minimizing Authentication Data
- UA-7. Maintenance of Authenticated User Status
- UA-8. Support for Multiple User-Authentication Mechanisms

Component IA_UA.5. Multiple Authentication

This component extends IA_UA.4 by including a requirement for multiple user authentication. It consists of the following elements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data
- UA-4. Handling of Repeated Authentication Failures
- UA-5. Minimizing Exposure of Authentication Data
- UA-6. Minimizing Authentication Data
- UA-7. Maintenance of Authenticated User Status
- UA-8. Support for Multiple User-Authentication Mechanisms

- UA-9. Multiple User Authentication

Component IA_UA.6. Single Login to Distributed Systems

This component extends IA_UA.3 by including a requirement for single login to distributed systems. It consists of the following elements:

- UA-1. User-Authentication Support
- UA-2. User-Authentication Data
- UA-3. Protection of User-Authentication Data
- UA-4. Handling of Repeated Authentication Failures
- UA-5. Minimizing Exposure of Authentication Data
- UA-6. Minimizing Authentication Data
- UA-7. Maintenance of Authenticated User Status
- UA-10. Single-User Login

It is envisioned that component IA_UA.1 will be used in profiles for systems with limited capabilities, such as automated guards, where minimal user I&A are the primary functions supported. Component IA_UA.2 can be used in profiles for products where a defined user I&A policy is intended to complement access control, system entry, and availability policies. To this end, IA_UA.2 offers some requirements for protection against penetration attempts via the login mechanism. Component IA_UA.3 strengthens the protection of the authentication data primarily in distributed systems. However, its use is envisioned in the same types of profiles as those of IA_UA.2. Component IA_UA.4 extends the feature coverage of IA_UA.3 by requiring system support for separate user authentication mechanisms for specific policy attributes, and, as such, use of this component is anticipated in profiles for systems with advanced access control and system entry policies. Component IA_UA.5 can strengthen user authentication by requiring use of multiple authentication procedures and, thus, it is anticipated that IA_UA.5 will be used in profiles for high-security systems. Component IA_UA.6 is intended for use in distributed system profiles where the protection of user-authentication data is further enhanced by requiring single user logins (as opposed to separate, per-host logins).

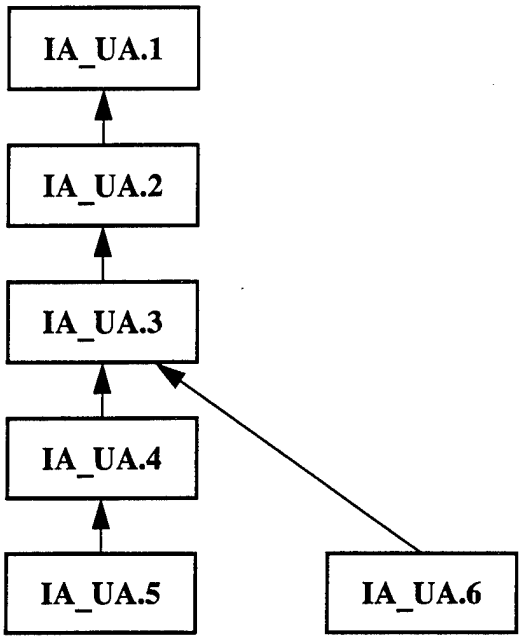


Figure 12. Component Relationships: User Authentication

4. INTER-REALM AUTHENTICATION

4.1 ELEMENTS

IRA-1. Defined Authentication Paths

(a) If a distributed system is partitioned into separate administrative realms, authentication paths among the authorities of multiple realms shall be defined and enforced in accordance with the defined policy for inter-realm authentication.

(b) In this case, a subject shall be able to discover the identity of the authorities that have been trusted in order to authenticate another subject (i.e., the authorities of an authentication path).

Dependencies:

- Uses: LP-1, CA-1, CA-2, CA-5–CA-8

IRA-2. Authentication of Traveling Users

(a) In a multi-realm distributed system, a traveling user shall be able to authenticate (e.g., login) in a foreign realm in accordance with the inter-realm authentication policy.

Dependencies:

- Uses: IRA-1

IRA-3. Avoiding Shared Secrets

(a) The authentication function shall avoid the use of shared secrets for inter-realm authentication.

Dependencies:

- Uses: IRA-1

4.2 COMPONENTS

The components defined below are intended for use in distributed systems where the TCBs used in authentication belong to different administrative realms, and where users can travel and initiate authentication in different realms. For this reason, authentication paths need to be defined among the authentication authorities of these realms. In such distributed systems, use of shared secrets for authentication may require that secure authentication authorities (e.g., servers) be on-line continuously to perform channel and user authentication. Use of on-line authorities is less desirable than use of off-line authorities. For this reason it is recommended that shared secrets, which require on-line authorities, should be avoided whenever possible. The components of this family are rated based on the feature coverage and strength of the included elements.

Component IA_IRA.1. Defined Authentication Paths

This component requires the definition of authentication paths. It consists of the following element:

- IRA-1. Defined Authentication Paths

Component IA_IRA.2. Authentication of Traveling Users

This component extends IA_IRA.1 by requiring support for travelling users, and consists of the following elements:

- IRA-1. Defined Authentication Paths
- IRA-2. Authentication of Traveling Users

Component IA_IRA.3. Enhanced Inter-Realm Authentication

This component strengthens the security of inter-realm authentication by avoiding the use of shared secrets. It consists of the following elements:

- IRA-1. Defined Authentication Paths
- IRA-2. Authentication of Traveling Users
- IRA-3. Avoiding Shared Secrets

It is envisioned that components IA_IRA.1 will be used in all profiles where inter-realm authentication for both channels and users is needed, whereas IA_IRA.2 will be used for profiles of requiring user mobility support. In contrast, IA_IRA.3 can be used in profiles where the strength of the authentication function needs to be increased (e.g., by relying on secure off-line, rather than on-line, servers).

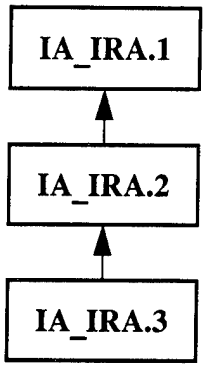


Figure 13. Component Relationships: Inter-Realm Authentication

5. AUTHENTICATION POLICY

5.1 ELEMENTS

IAP-1. Authentication Policy

(a) The authentication policy, or policies, supported by the authentication functions shall be defined and enforced.
(b) Each policy shall specify the types of subject and types of authentication supported (e.g., types of channels and user authentication).
(c) Each policy shall specify the time and duration of channel authentication (e.g., login session, a remote procedure call (RPC) bind, call, and packet authentication).
(d) Each policy shall specify the authentication revocation conditions.
(e) Each policy shall specify the validity and renewability of authentication data.
(f) Each policy shall specify the user-authentication mechanisms whenever multiple mechanisms are supported.
(g) Each policy shall specify the handling of the authentication failures.
(h) Each policy shall specify whether single login shall be supported.

IAP-1A. Inter-Realm Authentication Policy

(a) The authentication policy, or policies, supported by the authentication functions shall be defined and enforced.
(b) Each policy shall specify the types of subject and types of authentication supported (e.g., types of channels and user authentication).
(c) Each policy shall specify the time and duration of channel authentication (e.g., login session, remote procedure call (RPC) bind, call, and packet authentication).
(d) Each policy shall specify the authentication revocation conditions.
(e) Each policy shall specify the validity and renewability of authentication data.
(f) Each policy shall specify the user-authentication mechanisms whenever multiple mechanisms are supported.
(g) Each policy shall specify the handling of the authentication failures.
(h) Each policy shall specify whether single login shall be supported.
(i) Each policy shall specify the authentication path (i.e., which of the TCBs and trusted authentication authorities of the distributed system are used to perform authentication).

5.2 COMPONENTS

The two components below require the definition of an authentication policy applicable to distributed systems. These components are rated based on the feature coverage of the authentication-policy definition elements.

Component IA_IAP.1. Authentication Policy Definition

This component provides a minimal set of distributed authentication policy requirements. It consists of the following element:

- IAP-1. Authentication Policy

Component IA_IAP.1A. Inter-Realm Authentication Policy Definition

This component addresses distributed authentication policy requirements for multiple-realm operational environments. It consists of the following element:

- IAP-1A. Inter-Realm Authentication Policy

It is envisioned that component IA_IAP.1 will be used in all profiles requiring distributed authentication policy definition, while component IA_IAP.1A will provide the additional requirements for those profiles requiring multiple realm support.

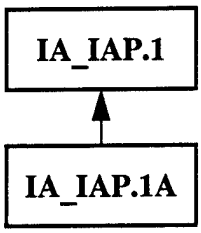


Figure 14. Component Relationships: Authentication Policy

C. SYSTEM ENTRY CLASS

Families:

1. Distributed System Entry

System entry specifies functional requirements for the control of an identified and authenticated user's entry into the system. The user's entry into the system typically consists of the creation of one or more subjects that execute instructions in the system on behalf of the user. At the end of the system entry procedure, provided the system entry conditions are satisfied, the created subjects bear the policy attributes determined by the identification and authentication (I&A) functions. System entry conditions can be specified in terms of policy attributes such as the user's identity, group or role membership, confidentiality and integrity levels, time intervals, location, and mode of access.

The system entry procedure may include warnings about unauthorized attempts to gain access to the system. It may also display last login data to the user, so that the user can determine whether the previous successful login was performed by the user and not by an intruder who successfully broke the user's password, for instance. The system entry procedure may enable control over (1) multiple simultaneous user logins, (2) the locking of an interactive session during periods of user inactivity, (3) time intervals for authorized user access, and (4) location or port of user entry.

System entry control can help counter threats of inadvertent, deliberate, or coerced access performed in an unauthorized manner by an authenticated user. For example, the location and time of system entry can be constrained in such a way that identified and authenticated users located in areas of high exposure (e.g., public areas) cannot display sensitive data, enter high-integrity commands, or operate outside working hours. Similarly, controlling the mode of system entry helps ensure that identified and authenticated users cannot remotely start batch computations that would normally require the user's attendance.

1. DISTRIBUTED SYSTEM ENTRY

1.1 ELEMENTS

DSE-1. Warning Message Display

(a) Prior to initiating the system login procedure, the TCB shall display an advisory warning message to the user regarding unauthorized use of the system and the possible consequences of failure to heed this warning.

DSE-2. Prior Authentication

(a) Before system entry is granted to a user, the identity of that user shall be authenticated by the TCB.

(b) If the TCB is designed to support multiple login sessions per user identity, the TCB shall provide a protected mechanism to enable limiting the number of login sessions per user identity or account with a default of a single login session.

Dependencies:

- Uses: UA-1

DSE-3. System Entry Conditions

(a) The TCB shall grant system entry only in accordance with the authenticated user's policy attributes.

(b) The system entry conditions shall be expressed in terms of users' policy attributes (e.g., greatest lower bound and least upper bound computations including the user levels, terminal levels, system levels).

(c) If no explicit system entry conditions are defined, the system entry default shall be used (e.g., the correct user authentication).

Dependencies:

- Uses: DSE-2, UA-1, SPS-2, MPA-1, MPA-3, MPA-3A
- Policy: ACA-1, ACA-2, ACA-6

DSE-4. Location-Based System Entry Control

(a) The TCB shall provide a protected mechanism to allow or deny system entry

based on location or port of entry.

(b) Conditions for system entry via dial-up lines (e.g., lists of user identities authorized to enter the system via dial-up lines), if any, shall be specified.

Dependencies:

- Uses: DSE-3
- Policy: DSE-3

DSE-5. Time-Based System Entry Control

(a) The TCB shall provide a protected mechanism to allow or deny system entry based on specified ranges of time.

(b) Entry conditions using these ranges shall be specified using time-of-day, day-of-week, and calendar dates.

Dependencies:

- Uses: DSE-3
- Policy: DSE-3

DSE-6. Display and Modification of System Entry Attributes

(a) The TCB shall provide a protected mechanism that enables authorized administrators to display and modify the policy attributes used in system entry control for each user.

(b) The conditions under which an unprivileged user may display these attributes shall be specified.

Dependencies:

- Uses: DSE-2, MPA-1, MPA-3, MPA-3A

DSE-7. Display of User's Entry Data

(a) Upon a user's successful entry to the system, the TCB shall display the following data to the user and shall not remove them without user intervention:

1. The date, time, means of access and port of entry of the last successful entry to the system; and
2. The number of successive unsuccessful attempts to access the system since the last successful entry by the identified user.

Dependencies:

- Uses: DSE-2

DSE-8. User Inactivity Handling

(a) The TCB shall either lock or terminate an interactive session after an admin-

istrator-specified interval of user inactivity.

(b) The default value for this interval shall be specified.

Dependencies:

- Uses: DSE-2, MPA-3, MPA-3A

DSE-8A. User-Initiated Inactivity Handling

(a) The TCB shall either lock or terminate an interactive session after an administrator-specified interval of user inactivity.

(b) The default value for this interval shall be specified.

(c) The TCB shall also provide a mechanism for user-initiated locking of the user's own interactive sessions (e.g., keyboard locking) that includes the following:

1. **Clearing or over-writing display devices to make the current contents unreadable;**
2. **Requiring user authentication prior to unlocking the session; and**
3. **Disabling any activity of the user's data entry and display devices other than unlocking the session.**

Dependencies:

- Uses: DSE-2, MPA-3, MPA-3A

1.2 COMPONENTS

All component of this family share five types of system entry requirements, namely (1) the warning message display, (2) the authentication of users prior to system entry, (3) the definition of the system entry conditions (i.e., system entry policy), (4) the ability to display and modify the initial system entry attributes in an authorized manner, and (5) the ability to display the entry attributes for users that successfully performed the system entry procedure. Additional requirements of user inactivity handling are included in some of the components. The components provided below illustrate the use of the five requirement types, and are rated based on the coverage of the elements in each component.

Component SE_DSE.1. Basic System Entry Control

This component contains the basic, common requirements of system entry. It consists of the following elements:

- DSE-1. Warning Message Display
- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data

Component SE_DSE.2. System Entry and Session Control

This component extends SE_DSE.1 by including a requirement for user-inactivity handling. It consists of the following elements:

- DSE-1. Warning Message Display
- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data
- DSE-8. User Inactivity Handling

Component SE_DSE.2A. System Entry and User's Session Control

This component extends SE_DSE.2 by including a requirement for allowing the user to initiate the locking of his or her own interactive session. It consists of the following elements:

- DSE-1. Warning Message Display

- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data
- DSE-8A. User-Initiated Inactivity Handling

Component SE_DSE.3. Location-Based System Entry and User's Session Control

This component extends SE_DSE.2A by requiring that the system entry conditions include location-based entry conditions. It consists of the following elements:

- DSE-1. Warning Message Display
- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-4. Location-Based System Entry Control
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data
- DSE-8A. User-Initiated Inactivity Handling

Component SE_DSE.4. Time-Based System Entry and User's Session Control

This component extends SE_DSE.3 by requiring that the system entry conditions include time-based entry conditions. It consists of the following elements:

- DSE-1. Warning Message Display
- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-5. Time-Based System Entry Control
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data
- DSE-8A. User-Initiated Inactivity Handling

Component SE_DSE.5. Location- and Time-Based System Entry and User's Session Control

This component extends SE_DSE.3 and SE_DSE.4 by requiring that the system entry conditions include both time- and location-based entry conditions. It consists of the following elements:

- DSE-1. Warning Message Display

- DSE-2. Prior Authentication
- DSE-3. System Entry Conditions
- DSE-4. Location-Based System Entry Control
- DSE-5. Time-Based System Entry Control
- DSE-6. Display and Modification of System Entry Attributes Conditions
- DSE-7. Display of User's Entry Data
- DSE-8A. User-Initiated Inactivity Handling

We anticipate that the first three components, SE_DSE.1, SE_DSE.2, and SE_DSE.2A, will be used in the majority of profiles where the time and location of entry are inconsequential to the system entry policy. The last three components can be used in profiles where the system entry policy requires location and time control, and where access to sensitive data is based on the location and time of entry.

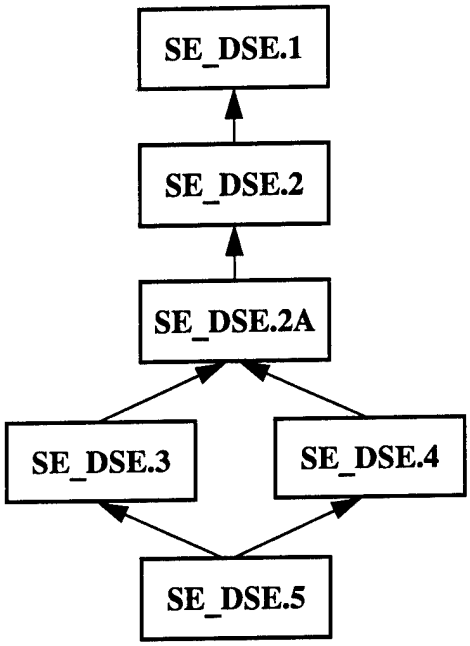


Figure 15. Component Relationships: Distributed System Entry

D. TRUSTED PATH CLASS

Families:

1. Distributed Trusted Path

Trusted path specifies functional requirements for ensuring that users have direct, unencumbered communication with a local or remote-host TCB. For distributed systems, a trusted path mechanism is implemented with trusted channels. A trusted path may be required at login time and at other times during a subject's session. Trusted path exchanges may be initiated by a user during an interaction with a local- or remote-host TCB. However, a TCB or a trusted-application request for user input should also allow a user to initiate communication and respond via the trusted path. A user's response via the trusted path guarantees that active intruders or untrusted applications cannot intercept and/or modify the user's response.

The threats countered by the trusted-path requirements are unauthorized discovery and/or modification of user-private information associated with commands (e.g., login password, sensitivity of the user's actions), and modification of commands and command parameters causing incorrect user input to a host TCB. Trusted-path programs of a host's TCB may also be invoked by trusted applications to ensure correct display of information to the user. These programs may also allow the addition of trusted application commands to the trusted path so that users could communicate securely with these applications.

Absence of a trusted path may allow breaches of accountability in environments where untrusted applications are used. These applications can intercept user-private information, such as passwords, and use it to impersonate other legitimate users. As a consequence, responsibility for any system actions cannot be reliably assigned to an accountable entity. Also, these applications could output erroneous information on an unsuspecting user's display. Thus, subsequent user actions may be erroneous and may lead to security breaches.

1. DISTRIBUTED TRUSTED PATH

1.1 ELEMENTS

DTP-1. Login Trusted Channel

(a) A trusted channel between a user and a local or remote-host TCB for initial identification and authentication shall be supported by each host TCB.
(b) This channel shall maintain the confidentiality and integrity of the user identification and authentication attributes and of the local or remote-host TCB reply.
(c) This channel shall also enable the user to unambiguously establish the identity of the local or remote-host TCB (e.g., certified login).
(d) The user shall be able to receive legitimate (e.g., not modified, substituted, nor replayed) confirmation of identification and authentication from the local or remote-host TCB.

Dependencies:

- Uses: LP-1, DCF-1–DCF-8, DIF-1–DIF-8, IAI-1–IAI-2

DTP-1A. Trusted Channel for User Communications

(a) A trusted channel between a user and a local or remote-host TCB shall be supported by each host TCB.
(b) This channel shall **provide identification and authentication, confidentiality, and integrity of all user-to-TCB communication (e.g., command and data messages, and message streams or sequences).**
(c) This channel shall also enable the user to unambiguously establish the identity of the local or remote-host TCB (e.g., certified login).
(d) The user shall be able to receive legitimate (e.g., not modified, substituted, nor replayed) confirmation of identification and authentication from the local or remote-host TCB.
(e) TCB commands supported via the trusted channel shall use confidentiality or integrity protection as necessary for all user-TCB communication.
(f) If the trusted channel must cross multiple realms, the authentication path necessary to establish the trusted channel shall include only realms that are trust-

ed by both the user's local realm and by the remote-TCB realm.

Dependencies:

- Uses: LP-1, DCF-1–DCF-8, DIF-1–DIF-8, IAI-1–IAI-2, IRA-1

DTP-2. Trusted Channels for Mobile Users

(a) Trusted channels shall be supported for mobile (e.g., traveling) users in realms other than those where the user are registered.

Dependencies:

- Uses: LP-1, DTP-1A

DTP-3. Trusted Application-to-User Channels

(a) Both a local and a remote-host TCB shall be capable of establishing a trusted channel between its trusted applications and users whenever trusted application-to-user communication is required (e.g., display or input of valued or sensitive application data).

Dependencies:

- Uses: LP-1, DTP-1A

1.2 COMPONENTS

Four trusted channel components presented below combine three independent trusted channel elements, namely trusted channels for user communications, trusted channels for mobile users, and trusted application-to-user communication. Another component, addressing only trusted login, is also included. Other trusted channel components are possible. The trusted channel components defined below are rated based on scope and coverage of the elements in the components.

Component TP_DTP.1. Login Trusted Channel

This component is intended to cover the basic requirements for login trusted channel that are necessary for all profiles. This component consists of the following element:

- DTP-1. Login Trusted Channel

Component TP_DTP.1A. Trusted Channel for User Communications

This component is intended to cover all user-to-TCB communication. This component consists of the following element:

- DTP-1A. Trusted Channel for User Communications

Component TP_DTP.2. Trusted Channels for Mobile Users

This component extends TP_DTP.1A to cover all user-to-TCB communication for both stationary and mobile users. This component consists of the following elements:

- DTP-1A. Trusted Channel for User Communications
- DTP-2. Trusted Channels for Mobile Users

Component TP_DTP.3. Trusted Application-to-User Communication

This component extends TP_DTP.1A to cover both user-to-TCB communication and application-to-user communication. This component consists of the following elements:

- DTP-1A. Trusted Channel for User Communications
- DTP-3. Trusted Application-to-User Channels

Component TP_DTP.4. Trusted Communications for Mobile Users

This component extends TP_DTP.2A to cover both user-to-TCB communication and application-to-user communication for both stationary and mobile users. This component consists of the following elements:

- DTP-1A. Trusted Channel for User Communications
- DTP-2. Trusted Channels for Mobile Users
- DTP-3. Trusted Application-to-User Channels

The login trusted channel component, TP_DTP.1, is the most rudimentary form of trusted channel and, as such, it can be used in all profiles for environments where active login attacks are a threat. Component TP_DTP.1A can be used in profiles for environments where all user communication with the TCB can be subject to active attacks. Component TP_DTP.2 can be used in profiles intended to establish trusted channels for mobile user environments. Component TP_DTP.3 can be used in profiles for environments where sensitive application input from, or output to, the user must be protected. Component TP_DTP.4 combines the requirements of TP_DTP.2 and TP_DTP.3, thus providing the highest protection afforded by trusted channel. As such, this component can be used in profiles for high-security environments.

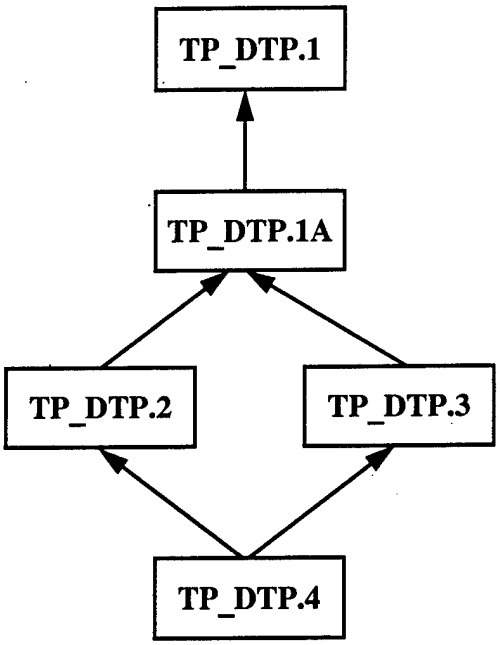


Figure 16. Component Relationships: Distributed Trusted Path

E. DATA CONFIDENTIALITY CLASS

Families:

1. **Data Confidentiality Functions**
2. **Data Confidentiality Policy**

Data confidentiality specifies requirements for policies and mechanisms whose goal is to ensure that sensitive data are not disclosed in an unauthorized manner while being transmitted between the host TCBs of a distributed system via communication channels.

Whenever the communication media are unprotected, encryption is required for data confidentiality. The use of encryption is governed by the requirements of the confidentiality policy, its supporting mechanisms, and strength of the confidentiality protection deemed necessary.

Confidentiality policy is intended to specify the scope of data confidentiality protection (e.g., what data items of a message must be protected) and the allowable leakage of data via confidential channels (i.e., the bypass rates). Also, when encryption must be used, the confidentiality policy specifies the types of cryptographic algorithms to be used and the context of use (e.g., per subject and per attribute basis), the modes of encryption allowed, if any, and the policy exemptions.

The strength of data confidentiality protection is also specified by the requirements of this class. These requirements specify that the channel physical protection shall be consistent with the overall protection policy (e.g., the TCB and channel physical protection). When encryption is used, strength requirements are specified independently of the assumed strength of the cryptographic function, which is already specified in the cryptographic support requirements. This is the case because, in practice, weak data-confidentiality functions can be designed with unbreakable cryptographic functions.

1. DATA CONFIDENTIALITY FUNCTIONS

1.1 ELEMENTS

DCF-1. Channel Protection

(a) Channels shall be protected by physical and administrative means.

(b) Physical protection shall ensure that compromise of data confidentiality is not feasible as a consequence of tampering with, or damage to, communication processors and media.

(c) The degree of physical and administrative protection of the communication processors and media assumed by the design of the data confidentiality functions shall be consistent with that assumed by the system security policy.

Dependencies:

- Uses: LP-1
- Policy: PP-2–PP-2B

DCF-2. Restricted Channel Routing

(a) It shall be possible to restrict the routing of channel data to use secure communication media (e.g., network links that are physically protected).

Dependencies:

- Uses: DCF-1

DCF-3. Channel Separation

(a) Data confidentiality functions shall have the capability to separate channels on the basis of selected policy attributes.

Dependencies:

- Uses: DCF-1, SPS-3, SPS-4, MPA-4, MPA-5, ACA-1, ACA-6, ACA-7

DCF-4. Data Confidentiality Protection

(a) Data transmitted through a channel may be read only by subjects authorized to use that channel (e.g., these data may be read only by the intended recipi-

ents).

DCF-5. Scope of Data Confidentiality Protection

(a) The distributed system shall have the capability to protect the confidentiality of individual messages (e.g., requests, replies, commands, and selected data) and selected control fields (e.g., sender and receiver identities, timestamp, sequence number fields) of a channel.

Dependencies:

- Uses: DCF-1

DCF-6. Cryptographic Function Support

(a) Whenever physical and administrative means provide insufficient channel protection, data confidentiality functions based on encryption shall also be provided for all channels and protocols specified by data confidentiality policy.

Dependencies:

- Uses: DCF-1, CDP-1, CDP-1A, CDP-1B, SKM-1–SKM-11
- Policy: SCF-1, SCF-2, SCF-3, SCF-5

DCF-7. Configurable Cryptographic Algorithms

(a) It shall be possible to configure the system such that the data confidentiality functions use different cryptographic algorithms for different protocols (e.g., mail or interprocess communication data).
(b) The modes of encryption supported by each cryptographic algorithm shall be appropriately used for each communication protocol and medium.
(c) The configuration of different cryptographic algorithms in the distributed system shall be performed by authorized individuals.

Dependencies:

- Uses: DCF-1
- Policy: SCF-5, SPS-7, MPA-8

DCF-7A. Selective Configuration of Cryptographic Algorithms

(a) It shall be possible to configure the system such that the data confidentiality functions use different cryptographic algorithms for different protocols (e.g., mail or interprocess communication data) **and for different policy attributes (e.g., classified or sensitive but unclassified data).**
(b) The modes of encryption supported by each cryptographic algorithm shall be appropriately used for each communication protocol and medium.
(c) The configuration of different cryptographic algorithms in the distributed system

shall be performed by authorized individuals.

Dependencies:

- Uses: DCF-1
- Policy: SCF-5, SPS-7, MPA-8, ACA-1, ACA-6, ACA-7

DCF-8. Controlled Use of the Cryptographic Functions

(a) It shall be possible to selectively allow the use of encryption for confidentiality protection (e.g., by system privileges assigned to subject policy attributes).
(b) It shall also be possible to mandate the use of encryption for confidentiality protection on the basis of selected subject policy attributes.
(c) Control over the use of encryption for confidentiality protection shall be exercised by authorized individuals.

Dependencies:

- Uses: DCF-1
- Policy: SCF-5, SPS-7, MPA-1, MPA-8

DCF-9. Integrity of Confidential Data

(a) If integrity of confidential data is required, specific measures shall be taken to provide data integrity (i.e., it shall not be assumed that data confidentiality measures also provide integrity).

Dependencies:

- Uses: DCF-1
- Policy: DIF-4, DIF-4A, DIP-1

2.2 COMPONENTS

The functional components of data confidentiality are separated from the policy components to emphasize the point that, while useful, general mechanisms can be required by a standard, the policy requirements should not be tied with requirements for any mechanism. This gives the profile designer the latitude of relying on policy components only or, alternatively, of introducing functional components within a profile. If both types of components are used, the inter-component dependencies must be analyzed when these components are selected for use in a profile.

A large number of components can be created using the functional elements of data confidentiality defined above. However, not all combinations of functional elements would make sense, and not all components would necessarily consist of unique elements. All meaningful components include a core of four basic elements that are necessary regardless of whether the channel is protected by physical and administrative means or by cryptographic means. These elements are channel protection, data confidentiality protection, scope of data confidentiality protection, and integrity of confidential data. Some functional elements, such as that of restricted channel routing, are used primarily when physical and administrative channel protection are employed, and are less relevant when channels are protected via cryptographic means. Other functional elements, such as channel separation, can be used in various components regardless of the channel protection means, yet are required in only some environments.

The components illustrated below reflect the fact that some systems will rely on physical and administrative controls to protect communication channels whereas others will rely on encryption. The former require the logical extension of each host's TCB to include channels, and thus their application is restricted to physically and administratively secure environments. The latter is more general in the sense that it does not make channels part of each host's TCB, and thus channels and data can pass through unprotected communication media and intermediate systems. The first two components consist of elements that do not require use of encryption. They are rated based on coverage of the elements in the components.

Component DC_DCF.1. Data Confidentiality with Physically Protected Channels

This component is intended to cover the basic elements of physical and administrative protection for communication channels. As such, these channels are routed only through physically secure media and intermediary systems. Therefore, data confidentiality

protection depends exclusively on the protection features of the TCB. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-2. Restricted Channel Routing
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection
- DCF-9. Integrity of Confidential Data

Component DC_DCF.2. Attribute-Based Data Confidentiality

This component includes all the elements of DC_DCF.1 and, in addition, requires that data confidentiality functions be applied selectively based on different policy attributes. For example, data whose sensitivity attributes differ will require use of separate channels. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-2. Restricted Channel Routing
- DCF-3. Channel Separation
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection
- DCF-9. Integrity of Confidential Data

It is envisioned that component DC_DCF.1 will be used in the majority of profiles where access control is based on discretionary policies, whereas component DC_DCF.2 will be predominantly used in profiles where access control is based on non-discretionary policies.

The remaining four components consist of elements that require encryption support for data confidentiality. Support for encryption ranges from basic functional support with minimal policy restrictions to components where selective configuration and use of encryption are controlled by administrative means and cryptographic policy. These components are rated based on coverage of the elements in the components.

Component DC_DCF.3. Basic Cryptographic Support for Data Confidentiality

This component is functionally equivalent with DC_DCF.1 except that it does not assume complete physical protection of communication channels and restricted channel routing. This component requires the use of encryption for channel protection and assumes

that basic encryption functions are added on to existing TCBs. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection
- DCF-6. Cryptographic Function Support
- DCF-9. Integrity of Confidential Data

Component DC_DCF.4. Configurable Cryptographic Support for Data Confidentiality

This component extends the requirements of DC_DCF.3 by including the capability of configuring different cryptographic algorithms for different protocols and applications. It also requires the necessary administrative features for such configurations. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection
- DCF-6. Cryptographic Function Support
- DCF-7. Configurable Cryptographic Algorithms
- DCF-9. Integrity of Confidential Data

Component DC_DCF.5. Attribute-Based Cryptographic Support for Data Confidentiality

This component is functionally equivalent with DC_DCF.2 except that it does not assume complete physical protection of communication channels and restricted channel routing. This component requires the use of encryption for channel protection, and assumes that, as in component DC_DCF.4, configurable cryptographic support is available. Unlike component DC_DCF.2, this component separates channels on the basis of different policy attributes using different cryptographic algorithms. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-3. Channel Separation
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection

- DCF-6. Cryptographic Function Support
- DCF-7A. Selective Configuration of Cryptographic Algorithms
- DCF-9. Integrity of Confidential Data

Component DC_DCF.6. Controlled Use of Cryptographic Support for Data Confidentiality

This component extends the requirements of DC_DCF.5 by requiring the capability of controlling the use of the different cryptographic algorithms for data confidentiality in different protocols and applications. This control ranges from selectively allowing to mandating the use of encryption for different protocols and applications. It also requires the necessary administrative control of the use of encryption for data confidentiality. This component consists of the following elements:

- DCF-1. Channel Protection
- DCF-3. Channel Separation
- DCF-4. Data Confidentiality Protection
- DCF-5. Scope of Data Confidentiality Protection
- DCF-6. Cryptographic Function Support
- DCF-7A. Selective Configuration of Cryptographic Algorithms
- DCF-8. Controlled Use of Cryptographic Functions
- DCF-9. Integrity of Confidential Data

It is envisioned that components DC_DCF.3 and DC_DCF.4 will be used in the majority of profiles where access control is based on discretionary policies, whereas components DC_DCF.5 and DC_DCF.6 will be predominantly used in profiles where access control is based on non-discretionary policies. Furthermore, component DC_DCF.6 can be used in environments where significant administrative control needs to be exercised over the use of encryption in various communication protocols and applications.

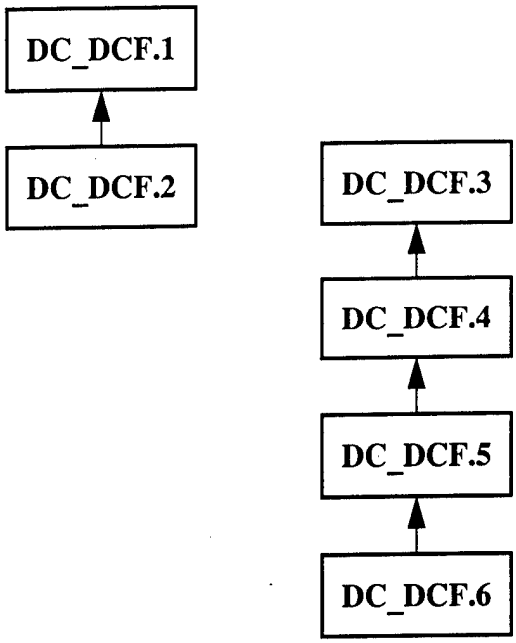


Figure 17. Component Relationships: Data Confidentiality Functions

2. DATA CONFIDENTIALITY POLICY

2.1 ELEMENTS

DCP-1. Definition and Enforcement of Data Confidentiality Policy

(a) The policy, or policies, supported by the data confidentiality functions of channels and protocols shall be defined and enforced.

Dependencies:

- Uses: DCF-1

DCP-2. Risk Thresholds for Protected Channels

(a) For each protected channel and protocol, the risk that any message or message stream can be disclosed in an unauthorized manner shall be less than a specified threshold.

Dependencies:

- Uses: DCP-1

DCP-3. Scope of Data Confidentiality Policy

(a) The data confidentiality policy shall define the scope of confidentiality protection. For each communication function of a channel and protocol, the data items and structures whose confidentiality is protected shall be identified.

Dependencies:

- Uses: DCP-1

DCP-4. Mandated Cryptographic Protection

(a) Whenever physical and administrative means provide insufficient channel protection, all sensitive data items shall be encrypted before transmission.

Dependencies:

- Uses: DCP-1, DCF-6

DCP-5. Cryptographic Policy Specification

(a) If based on encryption, the data confidentiality policy shall define the modes of encryption to be used.

(b) If based on encryption, the data confidentiality policy shall define the cryptographic algorithm to be used.

Dependencies:

- Uses: DCP-1, DCF-6
- Policy: SCF-1, SCF-2, SCF-3, SCF-5

DCP-5A. Cryptographic Policy Selectivity

(a) If based on encryption, the data confidentiality policy shall define the modes of encryption to be used.

(b) If based on encryption, the data confidentiality policy shall **select the cryptographic algorithm for each communication function (e.g., for each protocol).**

Dependencies:

- Uses: DCP-1, DCF-6
- Policy: SCF-1, SCF-2, SCF-3, SCF-5, SPS-7, MPA-8

DCP-6. Controlled Selectivity of Cryptographic Policy

(c) If based on encryption, the data confidentiality policy shall control the use of data confidentiality protection (i.e., whenever data confidentiality protection is allowed and whenever it is mandated).

Dependencies:

- Uses: DCP-1, DCF-6
- Policy: SPS-7, MPA-8

DCP-7. Confidentiality Policy Exemptions

(a) Any data item, structure, or protocol control information that is exempt from the data confidentiality policy shall be separated from the rest by system privileges.

Dependencies:

- Uses: DCP-1

DCP-8. Bypass Data

(a) The leakage of sensitive data via channel bypass data (e.g., protocol control information) shall not exceed a policy-specified threshold (i.e., the allowed

bypass rate).

(b) A threshold shall be specified and enforced for the communication protocols and channels supported by the distributed system.

Dependencies:

- Uses: DCP-1

2.2 COMPONENTS

The data confidentiality policy components illustrated below are intended to be used in the same types of profiles as those intended for the functional data confidentiality components. It should be noted that, as with the functional components, a sharp distinction is made between the use of these components in environments where physical and administrative measures are assumed to be the only means of protecting the communication channels and use of these components where encryption is also assumed to be available. In the latter case, the physical and administrative protection policy must be augmented by cryptographic policy. The first two policy components include elements that do not require encryption policy specification. They are rated based on coverage of policy elements.

Component DC_DCP.1. Basic Data Confidentiality Policy

This component is intended to cover the minimum requirements for data confidentiality policy. It consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-3. Scope of Data Confidentiality Policy

Component DC_DCP.2. Risk-Based Data Confidentiality Policy

This component includes the elements of DC_DCP.1 and, in addition, requires that the specification of the risk threshold for each channel, which is an important element in all data confidentiality specifications. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy

It is envisioned that component DC_DCP.1 will be used in the majority of profiles that do not require encryption policy specification, whereas component DC_DCP.2 will be used in profiles that include qualitative assessments of system vulnerabilities.

The remaining five components include elements of cryptographic policy. These elements parallel the functional elements that include use of encryption and, in addition, include specification of policy exemptions and channels bypass rates.

Component DC_DCP.3. Data Confidentiality Policy with Basic Cryptographic Policy

This component extends DC_DCP.2 by mandating the use of encryption and basic cryptographic policy specification. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy
- DCP-4. Mandated Cryptographic Protection
- DCP-5. Cryptographic Policy Specification

Component DC_DCP.3A. Data Confidentiality Policy with Configurable Cryptographic Algorithms

This component extends the requirements of DC_DCP.3 by specifying the selection of different cryptographic algorithms for different protocols and applications. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy
- DCP-4. Mandated Cryptographic Protection
- DCP-5A. Cryptographic Policy Selectivity

Component DC_DCP.4. Data Confidentiality Policy with Cryptographic Control

This component extends the requirements of DC_DCP.3A by specifying the control of different cryptographic algorithms for different protocols and applications. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy
- DCP-4. Mandated Cryptographic Protection
- DCP-5A. Cryptographic Policy Selectivity
- DCP-6. Controlled Selectivity of Cryptographic Policy

Component DC_DCP.5. Data Confidentiality Policy with Exemptions

This component extends the requirements of DC_DCP.4 by specifying control over confidentiality exemptions. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy

- DCP-4. Mandated Cryptographic Protection
- DCP-5A. Cryptographic Policy Selectivity
- DCP-6. Controlled Selectivity of Cryptographic Policy
- DCP-7. Confidentiality Policy Exemptions

Component DC_DCP.6. Data Confidentiality with Leakage Control

This component extends the requirements of DC_DCP.5 by specifying an upper bound for the leakage of sensitive data via channel bypass. This component consists of the following elements:

- DCP-1. Definition and Enforcement of Data Confidentiality Policy
- DCP-2. Risk Thresholds for Protected Channels
- DCP-3. Scope of Data Confidentiality Policy
- DCP-4. Mandated Cryptographic Protection
- DCP-5A. Cryptographic Policy Selectivity
- DCP-6. Controlled Selectivity of Cryptographic Policy
- DCP-7. Confidentiality Policy Exemptions
- DCP-8. Bypass Data

It is envisioned that components DC_DCP.3, DC_DCP.3A, and DC_DCP.4 will be used in profiles where cryptographic policy ranges from basic cryptographic support to controlled use of cryptographic algorithms. Component DC_DCP.5 can be used in profiles where significant control is necessary both over the use of cryptographic functions and over the use of other confidentiality functions. Finally, component DC_DCP.6 can be used in profiles whose access control require non-discretionary policies with information flow control. In such environments, the leakage of sensitive data via covert channels can be a significant policy concern.

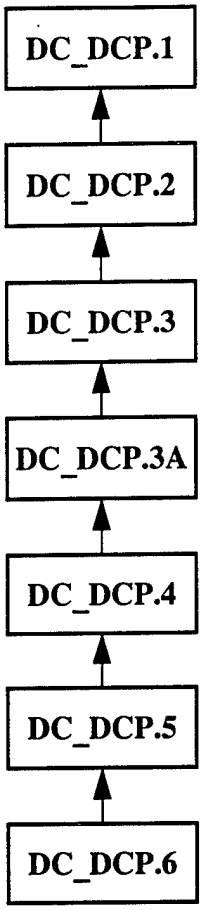


Figure 18. Component Relationships: Data Confidentiality Policy

F. DATA INTEGRITY CLASS

Families:

1. **Data Integrity Functions**
2. **Data Integrity Policy**

Data integrity specifies requirements for policies and mechanisms whose goal is to ensure that message data are not modified in an undetectable manner while being transmitted between the host TCBs of a distributed system via communication channels. Satisfying this goal also ensures that the source that created the message originally is unmodified and, therefore, becomes known to the message recipient.

Whenever the communication media is unprotected, cryptographic checksums or digital signatures are required for protecting data integrity. The use of encryption is governed by the requirements of data integrity policy, its supporting mechanisms, and strength of the integrity protection deemed necessary.

Integrity policy is intended to specify the scope of integrity protection (e.g., what data items of a message must be protected) and the integrity risk factors. Also, integrity policy is intended to allow configuration of integrity check functions, and to allow selection of integrity protection options by individual subjects. The integrity policy applies uniformly to both cryptographic and conventional integrity check functions.

The strength of data integrity protection is also specified by these requirements. As with data confidentiality, these requirements specify that the channel physical protection shall be consistent with the overall protection policy (e.g., the TCB and channel physical protection). When cryptographic integrity-check functions are used, strength requirements are specified independently of the assumed strength of the cryptographic function, which is already specified in the cryptographic support requirements. This is the case because, in practice, weak data-integrity functions can be designed with unbreakable cryptographic functions.

1. DATA INTEGRITY FUNCTIONS

1.1 ELEMENTS

DIF-1. Channel Protection

(a) Channels shall be protected by physical and administrative means.

(b) Physical protection shall ensure that compromise of data integrity is not feasible as a consequence of tampering with, or damage to, communication processors and media.

(c) The degree of physical and administrative protection of the communication processors and media assumed by the design of the data integrity functions shall be consistent with that assumed by the system security policy.

Dependencies:

- Uses: LP-1
- Policy: PP-2–PP-2B

DIF-2. Restricted Channel Routing

(a) It shall be possible to restrict the routing of channel data to secure communication media (e.g., network links that are physically protected).

Dependencies:

- Uses: DIF-1

DIF-3. Channel Separation

(a) Data integrity functions shall have the capability to separate channels on the basis of selected policy attributes.

Dependencies:

- Uses: DIF-1, SPS-3, SPS-4, MPA-4, MPA-5, ACA-1, ACA-6, ACA-7

DIF-4. Data Integrity Protection

(a) Data transmitted on a channel shall be protected by integrity check functions

as specified by the data integrity policy.

(b) The integrity check functions shall allow the detection of the following:

1. **Modification and substitution of a data item of a message or of a message stream (i.e., it shall be possible to determine that the data items of a message or message stream belong to that message or message stream),**
2. **Change in the order of a data item in a message or of a message stream (i.e., it shall be possible to determine that the data items of a message or message stream are correctly ordered in that message or message stream), and**
3. **Change in the length (i.e., number of data items) of a message or message stream (i.e., it shall be possible to determine that all data items of a message or message stream are included in that message or message stream).**

Dependencies:

- Uses: DIF-1

DIF-4A. Configurable Data Integrity Protection

(a) Data transmitted on a channel shall be protected by integrity check functions as specified by the data integrity policy.

(b) The integrity check functions shall allow the detection of the following:

1. Modification and substitution of a data item of a message or of a message stream (i.e., it shall be possible to determine that the data items of a message or message stream belong to that message or message stream),
2. Change in the order of a data item in a message or of a message stream (i.e., it shall be possible to determine that the data items of a message or message stream are correctly ordered in that message or message stream), and
3. Change in the length (i.e., number of data items) of a message or message stream (i.e., it shall be possible to determine that all data items of a message or message stream are included in that message or message stream).

(c) Data integrity protection shall have the capability to configure different integrity-check functions for different channels and protocols.

Dependencies:

- Uses: DIF-1

DIF-5. Scope of Data Integrity Protection

(a) For each communication channel and protocol, the data items and structures whose integrity is protected shall be identified.

(b) The distributed system shall have the capability to protect the integrity of individual messages (e.g., requests, replies, commands, and selected data) and selected control fields (e.g., headers, timestamps, sequence numbers, and ran-

dom-number fields).

Dependencies:

- Uses: DIF-1

DIF-5A. Data Integrity Protection with Replay Detection

(a) For each communication channel and protocol, the data items and structures whose integrity is protected shall be identified.

(b) The distributed system shall have the capability to protect the integrity of individual messages (e.g., requests, replies, commands, and selected data) and selected control fields (e.g., headers, timestamps, sequence numbers, and random-number fields).

(c) Replay detection functions (e.g., functions based on sliding time windows and replay buffers, sequence numbers, random numbers, or combinations thereof) shall allow the detection of replays of an old message, a message stream, or parts thereof.

Dependencies:

- Uses: DIF-1

DIF-5B. Data Integrity Protection with Message Sequencing

(a) For each communication channel and protocol, the data items and structures whose integrity is protected shall be identified.

(b) The distributed system shall have the capability to protect the integrity of individual messages (e.g., requests, replies, commands, and selected data) and selected control fields (e.g., headers, timestamps, sequence numbers, and random-number fields).

(c) Replay detection functions (e.g., functions based on sliding time windows and replay buffers, sequence numbers, random numbers, or combinations thereof) shall allow the detection of replays of an old message, a message stream, or parts thereof.

(d) Whenever a communication protocol requires message sequencing, the distributed system shall also have the capability to protect the integrity of message streams and message sequences (e.g., connection set-up and request-response protocols) on each communication channel.

Dependencies:

- Uses: DIF-1

DIF-6. Cryptographic Function Support

(a) Whenever physical and administrative means provide insufficient channel protection, data integrity functions, based on cryptographic checksums or digital signatures, shall also be provided for all channels and protocols specified

by data integrity policy.

Dependencies:

- Uses: DIF-1, CDP-1, CDP-A, CDP-B, SKM-1–SKM-11
- Policy: SCF-1, SCF-2, SCF-4, SCF-5

DIF-6A. Robust Cryptographic Function

(a) Whenever physical and administrative means provide insufficient channel protection, data integrity functions, based on cryptographic checksums or digital signatures, shall also be provided for all channels and protocols specified by data integrity policy.

(b) Cryptographic checksums and digital signatures shall ensure that the integrity policy can be preserved over the lifetime of the secret or private keys used.

(c) In particular, without the knowledge of secret or private keys, it shall be computationally infeasible to perform the following:

1. **Derive a signature or checksum for a plaintext message, and**
2. **Derive a plaintext message for a signature or checksum.**

Dependencies:

- Uses: DIF-1
- Policy: SCF-1, SCF-2, SCF-4

DIF-7. Configurable Cryptographic Algorithms

(a) It shall be possible to configure the system such that the data integrity functions use different cryptographic checksums or signatures for different protocols (e.g., mail or interprocess communication data).

(b) The configuration of different cryptographic checksums or signatures in the distributed system shall be performed by authorized individuals.

Dependencies:

- Uses: DIF-1
- Policy: SCF-5, SPA-7, MPA-8

DIF-7A. Selective Configuration of Cryptographic Algorithms

(a) It shall be possible to configure the system such that the data integrity functions use different cryptographic checksums or signatures for different protocols (e.g., mail or interprocess communication data) **and for different policy attributes (e.g., critical or essential but unclassified data).**

(b) The configuration of different cryptographic checksums or signatures in the distrib-

uted system shall be performed by authorized individuals.

Dependencies:

- Uses: DIF-1
- Policy: SCF-5, SPA-7, MPA-8, ACA-1, ACA-6, ACA-7

DIF-8. Controlled Use of the Cryptographic Functions

(a) It shall be possible to selectively allow the use of cryptographic checksums or digital signatures for integrity protection (e.g., by system privileges assigned to subject policy attributes).

(b) It shall also be possible to mandate the use of cryptographic checksums or digital signatures for integrity protection on the basis of selected subject policy attributes.

(c) Control over the use of cryptographic checksums or digital signatures for integrity protection shall be exercised by authorized individuals.

Dependencies:

- Uses: DIF-1
- Policy: SCF-5, SPS-7, MPA-1, MPA-8

1.2 COMPONENTS

The functional components of data integrity are separated from the policy components to emphasize the point that, while useful and general mechanisms can be required by a standard, the policy requirements should not be tied with requirements for any mechanism. This gives the profile designer the latitude of relying on policy components only or, alternatively, of introducing functional components within a profile. If both types of components are used, the inter-component dependencies must be analyzed when these components are selected for use in a profile.

A large number of components can be created using the functional elements of data integrity defined above. However, not all combinations of functional elements would make sense, and not all components would necessarily consist of unique elements. All meaningful components include a core of four basic elements that are necessary regardless of whether the channel is protected by physical and administrative means or by cryptographic means. These elements are channel protection, data integrity protection, scope of data integrity protection, and integrity of confidential data. Some functional elements, such as restricted channel routing, are used primarily when employing physical and administrative channel protection, and are less relevant when channels are protected via cryptographic means. Other functional elements, such as channel separation, can be used in various components regardless of the channel protection means, yet are required in only some environments.

The components illustrated below reflect the fact that some systems will rely on physical and administrative controls to protect communication channels whereas others will rely on encryption. The former require the logical extension of each host's TCB to include channels and, thus, their application is restricted to physically and administratively secure environments. The latter is more general in the sense that it doers not make channels part of each host's TCB, and thus channels and data can pass through unprotected communication media and intermediate systems. The first five components consist of elements that do not require use of encryption. They are rated based on coverage of the elements in the components.

Component DI_DIF.1. Data Integrity with Physically Protected Channels

This component is intended to cover the basic elements of physical and administrative protection for communication channels. As such these channels are routed only through physically secure media and intermediary systems. Therefore, data integrity pro-

tection depends exclusively on the protection features of the TCB. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-2. Restricted Channel Routing
- DIF-4. Data Integrity Protection
- DIF-5. Scope of Data Integrity Protection

Component DI_DIF.1A. Data Integrity with Replay Detection

This component extends DI_DIF.1 by adding a requirement for replay detection to basic data integrity protection. The addition of this requirement is important for environments where, despite the physical and administrative protection, message replays are possible. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-2. Restricted Channel Routing
- DIF-4. Data Integrity Protection
- DIF-5A. Data Integrity Protection with Replay Detection

Component DI_DIF.1B. Data Integrity with Message Sequencing

This component extends DI_DIF.1A by adding a requirement for data sequencing. The addition of this requirement is important for environments where, despite the physical and administrative protection, messages can be reordered by an attacker. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-2. Restricted Channel Routing
- DIF-4. Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing

Component DI_DIF.2. Attribute-Based Data Integrity

This component includes all the elements of DI_DIF.1B and, in addition, requires that data integrity functions be applied selectively based on different policy attributes. For example, data whose integrity attributes differ will require use of separate channels. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-2. Restricted Channel Routing

- DIF-3. Channel Separation
- DIF-4. Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing

Component DI_DIF.2A. Configurable Data Integrity Protection

This component includes all the elements of DI_DIF.2 and, in addition, requires that data integrity functions be applied selectively based on different policy attributes. For example, data whose integrity attributes differ will require use of separate channels and different integrity-check functions. This components consists of the following elements:

- DIF-1. Channel Protection
- DIF-2. Restricted Channel Routing
- DIF-3. Channel Separation
- DIF-4A. Configurable Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing

It is envisioned that component DI_DIF.1, DI_DIF.1A, and DI_DIF.1B will be used in the majority of profiles where access control is based on discretionary policies, whereas components DI_DIF.2, and DI_DIF.2A will be predominantly used in profiles where access control is based on non-discretionary policies.

The remaining five components consist of elements that require encryption support for data integrity. Support for encryption ranges from basic functional support with minimal policy restrictions to components where selective configuration and use of encryption are controlled by administrative means and cryptographic policy. These components are rated based on coverage and strength of the elements in the components.

Component DI_DIF.3. Basic Cryptographic Support for Data Integrity

This component is functionally equivalent with DI_DIF.1B except that it does not assume complete physical protection of communication channels and restricted channel routing. This component requires the use of encryption for channel protection and assumes that basic encryption functions are added on to existing TCBs. This components consists of the following elements:

- DIF-1. Channel Protection
- DIF-4. Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing

- DIF-6. Cryptographic Function Support

Component DI_DIF.3A. Robust Cryptographic Support for Data Integrity

This component extends the requirements of DI_DIF.3 by increasing the strength of the cryptographic support. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-4. Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing
- DIF-6A. Robust Cryptographic Function

Component DI_DIF.4. Configurable Cryptographic Support for Data Integrity

This component extends the requirements of DI_DIF.3A by including the capability of configuring different integrity-check functions and cryptographic algorithms for different protocols and applications. It also requires the necessary administrative features for such configuration. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-4A. Configurable Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing
- DIF-6A. Robust Cryptographic Function
- DIF-7. Configurable Cryptographic Algorithms

Component DI_DIF.5. Attribute-Based Cryptographic Support for Data Integrity

This component is functionally equivalent with DI_DIF.2 except that it does not assume complete physical protection of communication channels and restricted channel routing. This component requires the use of encryption for channel protection, and assumes that, as in component DI_DIF.4, configurable cryptographic support is available. Unlike component DI_DIF.2, this component separates channels on the basis of different policy attributes using different cryptographic algorithms. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-3. Channel Separation
- DIF-4A. Configurable Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing
- DIF-6A. Robust Cryptographic Function

- DIF-7A. Selective Configuration of Cryptographic Algorithms

Component DI_DIF.6. Controlled Use of Cryptographic Support for Data Integrity

This component extends the requirements of DI_DIF.5 by requiring the capability of controlling the use of the different cryptographic algorithms for data integrity in different protocols and applications. This control ranges from selectively allowing to mandating the use of encryption for different protocols and applications. It also requires the necessary administrative control of the use of encryption for data integrity. This component consists of the following elements:

- DIF-1. Channel Protection
- DIF-3. Channel Separation
- DIF-4A. Configurable Data Integrity Protection
- DIF-5B. Data Integrity Protection with Message Sequencing
- DIF-6A. Robust Cryptographic Function
- DIF-7A. Selective Configuration of Cryptographic Algorithms
- DIF-8. Controlled Use of Cryptographic Functions

It is envisioned that components DI_DIF.3, DI_DIF.3A, and DI_DIF.4 will be used in the majority of profiles where access control is based on discretionary policies, whereas components DI_DIF.5 and DI_DIF.6 will be predominantly used in profiles where access control is based on non-discretionary policies. Furthermore, component DI_DIF.6 can be used in environments where significant administrative control needs to be exercised over the use of cryptographic checksums and signatures in various communication protocols and applications.

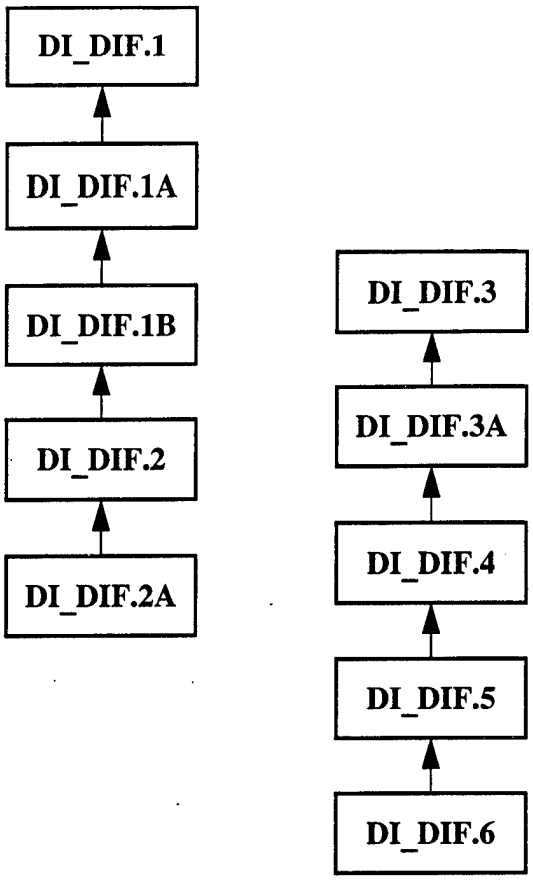


Figure 19. Component Relationships: Data Integrity Functions

2. DATA INTEGRITY POLICY

2.1 ELEMENTS

DIP-1. Definition and Enforcement of Data Integrity Policy

(a) The policy, or policies, supported by the data integrity functions of channels and protocols shall be defined and enforced.

Dependencies:

- Uses: DIF-1

DIP-2. Risk Thresholds for Protected Channels

(a) For each protected channel and protocol, the risk that any illegitimate (e.g., modified or replayed) message or message stream is accepted as legitimate by a recipient after the integrity check functions and replay detection functions have been employed is less than a specified threshold.

Dependencies:

- Uses: DIP-1

DIP-3. Scope of Data Integrity Policy

(a) The data integrity policy shall define the scope of integrity protection. For each communication function of a channel and protocol, the data items and structures whose integrity is protected shall be identified.

Dependencies:

- Uses: DIP-1

DIP-4. Mandated Cryptographic Protection

(a) Whenever physical and administrative means provide insufficient channel protection, cryptographic checksums or signatures shall be used before transmission.

Dependencies:

- Uses: DIP-1, DIF-6

DIP-5. Cryptographic Policy Specification

(a) If based on encryption, the data integrity policy shall define the modes of encryption to be used.

(b) If based on encryption, the data integrity policy shall define the cryptographic algorithm to be used.

Dependencies:

- Uses: DIP-1, DIF-6
- Policy: SCF-1, SCF-2, SCF-4, SCF-5

DIP-5A. Cryptographic Policy Selectivity

(a) If based on encryption, the data integrity policy shall define the modes of encryption to be used.

(b) If based on encryption, the data integrity policy shall **select the cryptographic checksum or signature algorithm for each communication function (e.g., for each protocol).**

Dependencies:

- Uses: DIP-1, DIF-6
- Policy: SCF-1, SCF-2, SCF-4, SCF-5, SPS-7, MPA-8

DIP-5B. Controlled Selectivity of Cryptographic Policy

(a) If based on encryption, the data integrity policy shall define the modes of encryption to be used.

(b) If based on encryption, the data integrity policy shall select the cryptographic checksum or signature algorithm for each communication function (e.g., for each protocol).

(c) If based on encryption, the data integrity policy shall control the use of data integrity protection (i.e., whenever data integrity protection is allowed and whenever it is mandated).

Dependencies:

- Uses: DIP-1, DIF-6
- Policy: SCF-1, SCF-2, SCF-5, SPS-7, MPA-8

2.2 COMPONENTS

The data integrity policy components illustrated below are intended to be used in the same types of profiles as those intended for the functional data integrity components. It should be noted that, as with the functional components, a sharp distinction is made between the use of these components in environments where physical and administrative measures are assumed to be the only means of protecting the communication channels and use of these components where encryption is also assumed to be available. In the latter case, the physical and administrative protection policy must be augmented by cryptographic policy. The first two policy components include elements that do not require encryption policy specification. They are rated based on coverage of policy elements.

Component DI_DIP.1. Basic Data Integrity Policy

This component is intended to cover the minimum requirements for data integrity policy. It consists of the following elements:

- DIP-1. Definition and Enforcement of Data Integrity Policy
- DIP-3. Scope of Data Integrity Policy

Component DI_DIP.2. Risk-Based Data Integrity Policy

This component includes the elements of DI_DIP.1 and, in addition, requires that the specification of the risk threshold for each channel, which is an important element in all data integrity specifications. This component consists of the following elements:

- DIP-1. Definition and Enforcement of Data Integrity Policy
- DIP-2. Risk Thresholds for Protected Channels
- DIP-3. Scope of Data Integrity Policy

It is envisioned that component DI_DIP.1 will be used in the majority of profiles that do not require encryption policy specification, whereas component DI_DIP.2 will be used in profiles that include qualitative assessments of system vulnerabilities. The remaining three components include elements of cryptographic policy. These elements parallel the functional elements that include use of encryption.

Component DI_DIP.3. Data Integrity Policy with Basic Cryptographic Policy

This component extends DI_DIP.2 by mandating the use of encryption and basic cryptographic policy specification. This component consists of the following elements:

- DIP-1. Definition and Enforcement of Data Integrity Policy

- DIP-2. Risk Thresholds for Protected Channels
- DIP-3. Scope of Data Integrity Policy
- DIP-4. Mandated Cryptographic Protection
- DIP-5. Cryptographic Policy Specification

Component DI_DIP.3A. Data Integrity Policy with Configurable Cryptographic Algorithms

This component extends the requirements of DI_DIP.3 by specifying the selection of different cryptographic algorithms for different protocols and applications. This component consists of the following elements:

- DIP-1. Definition and Enforcement of Data Integrity Policy
- DIP-2. Risk Thresholds for Protected Channels
- DIP-3. Scope of Data Integrity Policy
- DIP-4. Mandated Cryptographic Protection
- DIP-5A. Cryptographic Policy Selectivity

Component DI_DIP.4. Data Integrity Policy with Cryptographic Control

This component extends the requirements of DI_DIP.3A by specifying the control of different cryptographic algorithms for different protocols and applications. This component consists of the following elements:

- DIP-1. Definition and Enforcement of Data Integrity Policy
- DIP-2. Risk Thresholds for Protected Channels
- DIP-3. Scope of Data Integrity Policy
- DIP-4. Mandated Cryptographic Protection
- DIP-5A. Cryptographic Policy Selectivity
- DIP-6. Controlled Selectivity of Cryptographic Policy

It is envisioned that components DI_DIP.3, DI_DIP.3A, and DI_DIP.4 will be used in profiles where cryptographic policy ranges from basic cryptographic support to controlled use of cryptographic algorithms.

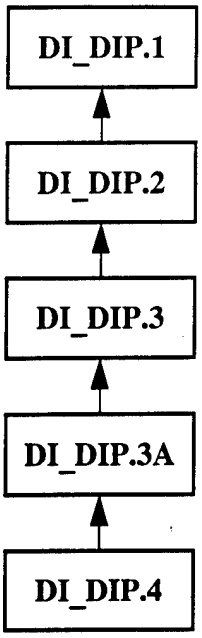


Figure 20. Component Relationships: Data Integrity Policy

G. CRYPTOGRAPHIC SUPPORT CLASS

Families:

1. **Secure Cryptographic Function**
2. **Cryptographic Domain Protection**
3. **Secure Key Management**

The cryptographic support requirements refer to both cryptographic mechanisms and policies. The goals of these requirements are the specification of a cryptographic function of appropriate strength and of algorithms to support it, the protection of the cryptographic domain, and the secure management of keys within a product.

The first goal is important because crypto-analytic attacks attempting to discover unknown secret keys used by these functions can be mounted against most functions of a product both by external intruders and by legitimate users. The importance of the cryptographic functions is increased by the fact that these functions are relied upon by several other components, such as identification and authentication, data confidentiality, and data integrity, which are the basis for secure-channel support in both centralized and distributed system products.

The second goal is important because the security of the cryptographic function can only be provided if the cryptographic domain is resistant to external interference and tampering when (1) executing the cryptographic algorithms in hardware, microcode, and/or software, (2) using the unknown secret key in plaintext form, and (3) maintaining configuration options, initialization data, and key storage. Breaches of cryptographic domain security would be particularly dangerous since they can potentially affect the security of all system users and TCBs beyond the boundaries of a single product host.

The third goal is important because the management of secret keys often provides the weakest link in the chain of cryptographic function mechanisms and use. Generation of poor-quality keys, inadequate key distribution, ineffective administrative procedures for key installation, weak key protection in storage, lack of limited key lifetime enforcement,

and incorrect separation of keys can lead to real security breaches by knowledgeable, determined attackers. (Numerous examples of such breaches in experimental systems, and also in production systems used in the commercial domain, can be found in the literature.)

1. SECURE CRYPTOGRAPHIC FUNCTION

1.1 ELEMENTS

SCF-1. Cryptographic Property Specification

(a) The cryptographic function shall satisfy the key-secrecy property. It shall also satisfy the cryptographic properties (i.e., secrecy, integrity, or both) required by the system security mechanisms and policies (e.g., identification and authentication, data confidentiality, data integrity).

Dependencies:

- Uses: LP-1

SCF-2. Key Secrecy Property

(a) The cryptographic function and its secret- or private-key space and lifetime shall be chosen so that the risk of unauthorized key discovery is within the threshold limit specified by the system security policy.

(b) During the lifetime of the cryptographically protected data, an exhaustive search that discovers the secret or private key shall be computationally infeasible.

Dependencies:

- Uses: SCF-1
- Policy: SKM-1, SKM-2, SKM-10, SKM-10A, SKM-10B, SKM-13, DIP-2, DCP-2.

SCF-3. Plaintext Secrecy Property

(a) The cryptographic function shall ensure that the mapping from ciphertext to plaintext is such that, given an element of ciphertext, the computation of the corresponding element of the plaintext is infeasible without knowing the secret or private key. (Such mappings can be provided both by symmetric and asymmetric cryptographic algorithms.)

Dependencies:

- Uses: SCF-1

SCF-4. Plaintext Integrity Property

(a) **The cryptographic function shall ensure that the mapping from plaintext to ciphertext is such that, given an element of plaintext, the computation of the corresponding element of the ciphertext is infeasible without knowing the secret or private key. (Such mappings can be provided both by symmetric and asymmetric cryptographic algorithms.)**

Dependencies:

- Uses: SCF-1

SCF-5. Cryptographic Algorithm Specification

(a) **Algorithms and operational modes used to implement the required cryptographic properties (i.e., key secrecy, plaintext secrecy, or plaintext integrity) shall be identified in the product specification.**

(b) **These algorithms shall be selected in accordance with international, national, and industry standards.**

Dependencies:

- Uses: SCF-1

1.2 COMPONENTS

Three distinct components can be created from the cryptographic functional elements. All components must include the requirements of cryptographic property specification and cryptographic algorithm specification. In particular, all components include the key secrecy property, without which sound key-based cryptographic functions cannot be supported. Also, all components include a requirement for cryptographic-algorithm selection in accordance with international, national, and industry standards. Thus, national cryptographic policies, if any, can be explicitly taken into account in the specification and implementation of a cryptographic algorithm. The rating of these components is based on the coverage of the cryptographic functional elements.

Component CR_SCF.1. Cryptographic Secrecy

This component is provided for use in profiles where the plaintext secrecy property must be used to implement confidentiality of system and application data (e.g., identification and authentication data, message confidentiality). This component consists of the following elements:

- SCF-1. Cryptographic Property Specification
- SCF-2. Key Secrecy Property
- SCF-3. Plaintext Secrecy Property
- SCF-5. Cryptographic Algorithm Specification

Component CR_SCF.2. Cryptographic Integrity

This component is provided for use in profiles where the plaintext integrity property must be used to implement data integrity mechanisms and policies (e.g., message authentication, file integrity). This component consists of the following elements:

- SCF-1. Cryptographic Property Specification
- SCF-2. Key Secrecy Property
- SCF-4. Plaintext Integrity Property
- SCF-5. Cryptographic Algorithm Specification

Component CR_SCF.3. Cryptographic Secrecy and Integrity

This component is provided for use in profiles where both the plaintext secrecy and integrity properties must be used to implement data secrecy integrity mechanisms and policies for various applications. This component consists of the following elements:

- SCF-1. Cryptographic Property Specification

- SCF-2. Key Secrecy Property
- SCF-3. Plaintext Secrecy Property
- SCF-4. Plaintext Integrity Property
- SCF-5. Cryptographic Algorithm Specification

It is envisioned that component CR_SCF.3 will be used in the vast majority of systems and applications, whereas components CR_SCF.1 and CR_SCF.2 will have more limited use, for example, in profiles for special applications and devices.

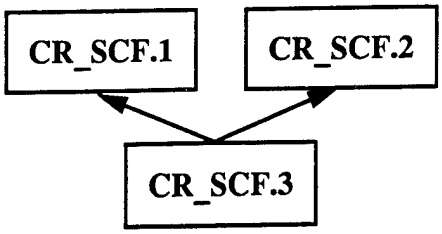


Figure 21. Component Relationships: Secure Cryptographic Function

2. CRYPTOGRAPHIC DOMAIN PROTECTION

2.1 ELEMENTS

CDP-1. Protection of Cryptographic Domain

(a) The cryptographic domain of each host shall be protected by the TCB of that host, and shall be non-circumventable and tamperproof.

Dependencies:

- Uses: LP-1, PP-2–PP-2B

CDP-1A. Logical Separation of Cryptographic Domain

(a) The cryptographic domain of each host shall be **a logically separate and distinct subset of the TCB domain of that host,** and shall be non-circumventable and tamperproof.

Dependencies:

- Uses: LP-1, PP-2–PP-2B

CDP-1B. Physical Separation of Cryptographic Domain

(a) The cryptographic domain of each host shall be a **physically** separate and distinct subset of the TCB domain of that host, and shall be non-circumventable and tamperproof.

Dependencies:

- Uses: LP-1, PP-2–PP-2B

CDP-2. Logical Protection of Cryptographic Keys

(a) The cryptographic domain shall ensure that the compromise (e.g., unauthorized disclosure, modification, substitution, or use) of secret or private keys is not possible as a consequence of using the cryptographic-domain functions.

(b) All covert channels of the cryptographic domain, if any, shall be handled in

accordance with the system security policy.

Dependencies:

- Uses: CDP-1, CDP-1A, CCH-1–CCH-3

CDP-3. Physical Protection of Cryptographic Keys

(a) Physical protection of the cryptographic domain shall ensure that compromise of secret or private keys is not possible as a consequence of physical tampering with, or damage to, the host.

(b) The degree of physical protection of the cryptographic domain shall be defined in accordance with the product security policy.

Dependencies:

- Uses: CDP-1B, PP-1, PP-2–PP-2B

CDP-4. Separation of Maintenance and Operational Modes

(a) If the cryptographic domain supports a maintenance mode, it shall clear all plaintext secret or private keys and other critical parameters when entering the maintenance mode, and shall clear all maintenance keys and other critical parameters when exiting the maintenance mode.

Dependencies:

- Uses: CDP-1, CDP-1A, CDP-1B

CDP-4A. Mandatory Separation of Maintenance and Operational Modes

(a) The cryptographic domain shall clear all plaintext secret or private keys and other critical parameters when entering the maintenance mode, and shall clear all maintenance keys and other critical parameters when exiting the maintenance mode.

Dependencies:

- Policy: CDP-1B, CDP-3

2.2 COMPONENTS

Three distinct components are created from the elements of cryptographic domain protection. All three components include requirements for protecting the cryptographic domain and secret keys, and for separating the maintenance and operational modes. The rating of the components is based on both the strength and coverage of the elements of cryptographic domain protection.

Component CR_CDP.1. Basic Cryptographic-Domain Protection

This component is intended to be used in profiles for systems where the cryptographic functions are included within each host's TCB as "add-on modules." The protection of the cryptographic functions is provided exclusively by the protection of each host's TCB, and the separation of maintenance and operational modes, if any, is provided by that of the host TCB. This component consists of the following elements:

- CDP-1. Protection of Cryptographic Domain
- CDP-4. Separation of Maintenance and Operational Modes

Component CR_CDP.2. Logical Separation of the Cryptographic Domain

This component is intended for use in profiles for systems where the cryptographic functions are integrated within a separate domain of each host's TCB. Additional protection of the cryptographic functions is provided to ensure that host TCB functions that are unprivileged with respect to the cryptographic domain cannot adversely affect the cryptographic-domain operation. As a consequence, the separation of maintenance and operational modes, if any, is provided separately from that of the host TCB. This component consists of the following elements:

- CDP-1A. Logical Separation of Cryptographic Domain
- CDP-2. Logical Protection of Cryptographic Keys
- CDP-4. Separation of Maintenance and Operational Modes

Component CR_CDP.3. Physical Separation of the Cryptographic Domain

This component is intended to be used in profiles for systems where the cryptographic functions are integrated within a separate, physically protected domain of each host's TCB. Both logical protection and physical protection of the cryptographic functions are provided to ensure that host TCB functions that are unprivileged with respect to the cryptographic domain cannot adversely affect the cryptographic-domain operation. Since the cryptographic domain is physically separated from host TCBs, the separation of main-

tenance and operational modes must be provided and must be separate from that of the host TCB. This component consists of the following elements:

- CDP-1B. Physical Separation of Cryptographic Domain
- CDP-2. Logical Protection of Cryptographic Keys
- CDP-3. Physical Protection of Cryptographic Keys
- CDP-4A. Mandatory Separation of Maintenance and Operational Modes

It is envisioned that, in most profiles for commercially available systems, CR_CDP.1 and CR_CDP.2 will be the predominant components. Component CR_CDP.1 provides a minimal set of requirements while component CR_CDP.2 will be used in profiles where structured protection mechanisms are required. The intent of component CR_CDP.2 is to recognize that compromising the cryptographic domain will result in security breaches beyond those of the local host TCB. For this reason, additional protection is necessary. Also, providing a separate domain specially tailored to the cryptographic function helps deny visibility to the internal states of the cryptographic operations (e.g., key generation). Such visibility could lead to cryptographic-function compromise. Component CR_CDP.3 is intended for use in highly protected environments where exposure of the cryptographic domain to physical security breaches is both possible and extremely damaging to the particular application.

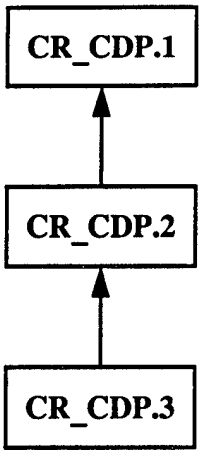


Figure 22. Component Relationships: Cryptographic Domain Protection

3. SECURE KEY MANAGEMENT

3.1 ELEMENTS

SKM-1. Secure Key Generation

(a) Key generation shall ensure that a secret or private key is unpredictable (e.g., the secret key has a random, secret value; derivation of the private key from publicly known or other secret keys is computationally infeasible).

(b) Intermediate key generation states and values shall not be accessible outside the cryptographic domain in plaintext or otherwise unprotected form.

(c) If a seed key is used for key generation, it shall be installed in the same manner that is used for the keys themselves.

Dependencies:

- Policy: SKM-2, SKM-10, SKM-10A, SKM-10B, SKM-13, DIP-2, DCP-2.
- Uses: CDP-1–CDP-4, CDP-4A

SKM-2. Key Attribute Specification

(a) A key shall consist of a random or pseudo-random key value and key attributes.

(b) A subject using a key shall be able to unambiguously identify the key attributes including the following:

1. **Type and identifier of the key,**
2. **Key version number (if any),**
3. **Date of key generation, and**
4. **Maximum lifetime of the key.**

Dependencies:

- Policy: SKM-1, SKM-10, SKM-10A, SKM-10B, SKM-13, DIP-2, DCP-2.
- Uses: CDP-1–CDP-4, CDP-4A

SKM-3. Key Installation

(a) Key installation into a host, which is a manual activity, shall be performed

using a protected function.

Dependencies:

- Uses: CDP-1

SKM-3A. Key Installation with Trusted Channel

(a) Key installation into a host, which is a manual activity, shall be performed using a protected function.

(b) The protected function shall require that a trusted channel be used for key installation (e.g., a TCB trusted channel, a smartcard-based trusted channel).

Dependencies:

- **Uses**: CDP-1, DTP-1, DTP-1A

SKM-3B. Protected Key Installation with Trusted Channel

(a) Key installation into a host, which is a manual activity, shall be performed using a protected function.

(b) The protected function shall require that a trusted channel be used for key installation (e.g., a TCB trusted channel, a smartcard-based trusted channel).

(c) The key shall be installed either in encrypted form or, using split-knowledge procedures, directly into the cryptographic domain (i.e., as two or more plaintext key components).

(d) When a key is installed under split-knowledge procedures, each individual entering each key component shall be authenticated.

Dependencies:

- **Uses:** CDP-1, CDP-1A, CDP-1B, DTP-1, DTP-1A, IAI-2, IAI-3

SKM-4. Closure of Installed-Key Use

(a) Key installation shall establish the set of subjects that is able to use the installed key (e.g., hosts, switches).

Dependencies:

- SKM-3, SKM-3A, SKM-3B

SKM-5. Key Distribution

(a) A capability for automatic key distribution among authenticated subjects shall be available.

(b) Key distribution shall perform the following:

1. **Maintain key protection, and**
2. **Establish the key is not an unauthorized replay and/or reuse.**

Dependencies:

- Uses: SCF-3, SCF-5

SKM-6. Closure of Distributed-Key Use

(a) Distributed-key distribution shall establish the set of subjects that is able to use the key.

Dependencies:

- Uses: SKM-5

SKM-7. Key Plaintext Protection

(a) All key values shall appear in plaintext form only within the cryptographic domain.

(b) The plaintext key values shall not be accessible from outside the cryptographic domain.

Dependencies:

- Uses: CDP-1–CDP-4

SKM-8. Key Storage Protection

(a) When not in use, a key shall be stored or archived in encrypted form in an area where it is protected from unauthorized disclosure, modification, substitution, or use.

Dependencies:

- Uses: LP-1, PP-2–PP-2B, SCF-3, SCF-5

SKM-9. Closure of Stored-Key Use

(a) Key storage and protection functions shall establish the set of subjects that is able to use the stored or archived key.

Dependencies:

- Uses: SKM-8

SKM-10. Key Destruction

(a) Protected functions shall be used to define and enforce limits (e.g., time or use limits) on key use and to extend these limits according to a well-defined policy.

Dependencies:

- Policy: SKM-1, SKM-2, SKM-13, DIP-2, DCP-2.

- Uses: CDP-1

SKM-10A. Key Destruction and Overwriting

(a) Protected functions shall be used to define and enforce limits (e.g., time or use limits) on key use and to extend these limits according to a well-defined policy.

(b) A capability shall be available to overwrite the plaintext keys, and unencrypted parameters within the cryptographic domain.

Dependencies:

- Policy: SKM-1, SKM-2, SKM-13, DIP-2, DCP-2.
- Uses: CDP-1A, CDP-1B, CDP-2

SKM-10B. Timely Key Destruction

(a) Protected functions shall be used to define and enforce limits (e.g., time or use limits) on key use and to extend these limits according to a well-defined policy.

(b) Protected functions shall be capable of destroying an expired secret or private key within an interval of time that is specified by an authorized system administrator.

Dependencies:

- Policy: SKM-1, SKM-2, SKM-13, DIP-2, DCP-2.
- Uses: CDP-1A, CDP-1B, CDP-2

SKM-11. Separation of Key Use

(a) The cryptographic domain shall be capable of separating key types based on key use.

(b) Separate, independent keys shall be defined for each type of cryptographic function. For example:

1. **For functions using symmetric keys, separate keys shall be used for data encryption and decryption, generation and verification of message authentication codes, key import and export (e.g., use of a key encrypting key to import and export a session key), and key translation into another key; and**
2. **For functions using asymmetric keys, separate keys shall be used for authentication, signature generation and verification, and encryption and decryption.**

Dependencies:

- Uses: CDP-1A, CDP-1B, CDP-2

SKM-12. Key Import and Export

(a) A protected function shall be used to import (load) or export (store) the key to or from the cryptographic domain.

(b) The import operation shall activate the key within the cryptographic domain (i.e., the plaintext value of the key shall be available only within the cryptographic domain).

(c) The key export operation shall encrypt all secret or private keys within the cryptographic domain before storing outside the cryptographic domain.

Dependencies:

- Uses: CDP-1A, CDP-1B, CDP-2, SKM-8, SKM-11

SKM-13. Key Escrow

(a) If key escrow is necessary, the distributed system shall define and enforce a key-escrow policy that shall specify the following:

1. **Selection of types of keys to be escrowed,**
2. **Key identification via system-global identifiers,**
3. **Binding of the escrowed key to the subjects using that key,**
4. **The escrow period,**
5. **The escrow authority, and**
6. **Procedures for accessing the encrypted (secret or private) key within the escrow facility.**

(b) Protected functions shall be used for key escrow.

(c) Protected functions shall not circumvent the other key-management requirements of the distributed system.

Dependencies:

- Policy: SKM-1, SKM-2, SKM-10, SKM-10A, SKM-10B, DIP-2, DCP-2.
- Uses: SCF-3–SCF-5, SKM-2

SKM-14. Key Activation

(a) Direct installation of a (secret or private) key into the cryptographic domain shall not automatically activate the key (i.e., make the key available for use within the cryptographic domain).

(b) A facility shall be provided to activate an installed key for a specific user.

(c) The default shall be that an installed key is disabled (i.e., the key cannot be used within the cryptographic domain until the user explicitly activates it).

(d) An activation mechanism ensures that the binding between the user identity and the key is maintained after installation. For example, a key stored in "split" form is activated by the user when he supplies the complementary split value; a key stored within a cryptographic token (e.g., smart card) may be

activated when a user inputs a personal identification number (PIN) to the token.

(e) An active key shall be deactivated in response to an event signaling that the user has terminated his use of the key (e.g., the user has removed a crypto-ignition key or removed a cryptographic token from its reader).

Dependencies:

- Uses: SKM-3B, IAI-2, IAI-3

3.2 COMPONENTS

Three components are created from the elements of secure key management. All three components include requirement elements of key attribute specification, secure key generation, key installation, key distribution, key maintenance (i.e., closure of key use, key storage and protection, key retrieval, and key destruction), and separation of key use. Additional elements, such as key import and export, key activation and key escrow, are included in some of the components to provide comprehensive coverage of key management functions. The rating of these components is based on both strength and coverage of secure key management elements included in the components.

Component CR_SKM.1. Basic Key Management

This component is intended to cover the basic key management requirements of a secure system. Although basic, the elements of this component cover most of the key management functions needed in a wide variety of environments, and apply equally to systems where the cryptographic functions are included as "add-on modules" and to systems where they are integrated within structured protection mechanisms. This component consists of the following elements:

- SKM-1. Secure Key Generation
- SKM-2. Key Attribute Specification
- SKM-3. Key Installation
- SKM-4. Closure of Installed-Key Use
- SKM-5. Key Distribution
- SKM-6. Closure of Distributed-Key Use
- SKM-7. Key Plaintext Protection
- SKM-8. Key Storage Protection
- SKM-9. Closure of Stored-Key Use
- SKM-10. Key Destruction
- SKM-11. Separation of Key Use

Component CR_SKM.2. Extended Key Management

This component is intended to be used in profiles where emphasis is placed on protected key installation and destruction within the cryptographic domain. These two areas have been identified as two of the most common sources of inadequate key management functions that can lead to breaches of key security. Key escrow is introduced as a condition-

al requirement for environments where cryptographic policy include the ability to recover keys used in an authorized, controlled manner. This component consists of the following elements:

- SKM-1. Secure Key Generation
- SKM-2. Key Attribute Specification
- SKM-3A. Key Installation with Trusted Channel
- SKM-4. Closure of Installed-Key Use
- SKM-5. Key Distribution
- SKM-6. Closure of Distributed-Key Use
- SKM-7. Key Plaintext Protection
- SKM-8. Key Storage Protection
- SKM-9. Closure of Stored-Key Use
- SKM-10A. Key Destruction and Overwriting
- SKM-11. Separation of Key Use
- SKM-12. Key Import and Export
- SKM-13. Key Escrow

Component CR_SKM.3. Advanced Key Management

This component strengthens the requirements of component CR_SKM.2 by including a requirement for protected key installation and a separate activation requirement for installed keys. By adding the protected key installation and activation requirements, this component helps decrease the vulnerability of insecure key installation and use. It also strengthens the key maintenance requirements by including a requirement for timely key destruction. This component is intended to be used in profiles where emphasis is placed on minimizing known vulnerabilities of key management. This component consists of the following elements:

- SKM-1. Secure Key Generation
- SKM-2. Key Attribute Specification
- SKM-3B. Protected Key Installation with Trusted Channel
- SKM-4. Closure of Installed-Key Use
- SKM-5.Key Distribution
- SKM-6. Closure of Distributed-Key Use
- SKM-7. Key Plaintext Protection

- SKM-8. Key Storage Protection
- SKM-9. Closure of Stored-Key Use
- SKM-10B. Timely Key Destruction
- SKM-11. Separation of Key Use
- SKM-12. Key Import and Export
- SKM-13. Key Escrow
- SKM-14. Key Activation

It is envisioned that component CR_SKM.1 will be used in the vast majority of profiles that include environments where a separate cryptographic domain, distinct from a host TCB, is not supported. Use of component CR_SKM.2 is envisioned in profiles requiring systems whose TCB is equipped with a separate cryptographic domain and trusted channel, whereas use of component CR_SKM.3 is envisioned in profiles for high-security environments.

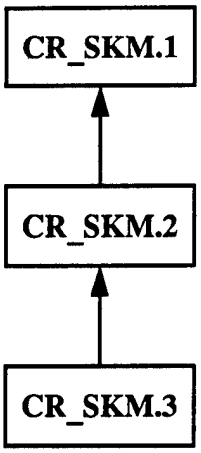


Figure 23. Component Relationships: Secure Key Management

H. ACCESS CONTROL CLASS

Families:[13]

1. **Definition of Access Control Attributes**
2. **Authorization of Subject Access to Objects**
3. **Administration of Access Control Attributes**

The access control objectives of organizational security policies can be divided into two types, namely confidentiality and integrity. These objectives determine whether the organization intends to prevent unauthorized disclosure or unauthorized modification and destruction of information. Often, organizational security policies include both confidentiality and integrity objectives to varying degrees. These policies reflect both security and system management goals that should be satisfied by multiple information technology (IT) products.

The extent to which an IT product's access control policy supports high-level system and organizational security policy objectives varies from product to product. Few commercial products are designed to support a single specific organizational policy. Instead, commercial products implement either low-level access control policies that can be tailored to support high-level organizational policies or multiple organizational policies that could be individually instantiated on a system basis. For example, some products implement both the Department of Defense (DoD) mandatory confidentiality policy (as modeled by Bell and LaPadula[14]) and a mandatory integrity policy (as modeled by Biba[15]). When using such IT products in environments where only the mandatory integrity policy needs to be enforced, the DoD mandatory confidentiality policy could be deconfigured (e.g., all authorization checks for DoD mandatory confidentiality would pass and all options for displaying or requesting confidentiality levels would be disabled). Similarly, other organizational

[13] The definition of access control policies using these families of elements is discussed in Appendix B.

[14] Bell, D. E. and LaPadula, L. J. October 1974. *Secure Computer Systems: Mathematical Foundations and Model.* M74-244. Bedford, MA: MITRE Corporation.

[15] Biba, K. J. April 1977. *Integrity Considerations for Secure Computer Systems.* ESD-TR-76-372. Bedford, MA: USAF Electronic Systems Division.

policies, such as the role-based access control policies, could be configured in a product when the environment of product use makes it necessary.

The access control policies in this section are IT product policies implemented by host TCBs and application servers. They are distinguished from the higher-level system and organizational security policies that generally use product policies to help achieve the higher-level security objectives. The access control elements are expressed as sets of policy-neutral requirements that can be assigned specific meaning in a variety of access control areas. For example, these requirements can be applied to gateway access control to define and enforce the security perimeter of a distributed system. They can also be applied to server access control to define and enforce server implemented policies, not just host TCB policies.

Product access control policies are designed to counter generic threats. These policies traditionally have been classified as discretionary or non-discretionary, depending upon whether the access control decisions regarding an object are primarily based either on actions of the unprivileged user and/or subject that created the object or on administrative actions. Access control policies of many products combine both discretionary and non-discretionary policies to counter different types of threats and eliminate various vulnerabilities.

1. DEFINITION OF ACCESS CONTROL ATTRIBUTES

1.1 ELEMENTS

ACA-1. Definition and Protection of Access Control Attributes

(a) Host TCBs shall define access control attributes for subjects (e.g., groups, roles, security levels) and objects (e.g., access rights, security levels).

(b) The access control attributes shall be protected from unauthorized modification and substitution.

(c) When transmitted across communication media, the confidentiality and integrity of the access control attributes shall be protected.

Dependencies:

- Uses: LP-1, PP-2–PP-2B, DCF-4, DCF-5, DIF-4, DIF-5
- Policy: IAI-1, IAI-1A, IAI-2, SE-3, DCP-1–DCP-5, DIP-1–DIP-5

ACA-2. Subject Attributes

(a) Subject attributes shall include the name of an individual, the defined groups and/or roles of which an individual is a member, or both.

(b) If the system is defined as a set of realms, the subject attributes shall include realm identifiers.

(c) If a security perimeter of the system is defined and enforced in the communication network (e.g., gateways, routers), the attributes that are used in access control decisions shall also include protocol control information, such as network identifiers, host identifier, protocol identifier, service identifier, and/or port identifier.

Dependencies:

- Uses: IAI-3, ACA-1
- Policy: IAI-1, IAI-1A, IAI-2

ACA-3. Object Attributes

(a) Object attributes shall include defined access rights (e.g., read, write, execute)

that can be assigned to subject attributes.

Dependencies:

- Uses: ACA-1

ACA-3A. Enhanced Object Attributes

(a) Object attributes shall include defined access rights (e.g., read, write, execute) that can be assigned to subject attributes.

(b) The access control attributes shall be capable of specifying for each object a list of subjects, simple or compound, and a list of groups and/or roles of subjects, with their respective access rights to that object.

(c) Furthermore, for each object, it shall be possible to specify a list of subjects, including groups and/or roles, for which no access to the object is given.

Dependencies:

- Uses: ACA-1

ACA-4. Device Attributes

(a) A host TCB shall support the assignment of access control attributes (e.g., device labels) for all attached physical devices.

(b) These attributes shall be used by the TCB to enforce constraints imposed by the physical environments in which the devices are located.

Dependencies:

- Uses: ACA-1

ACA-5. User Notification of Attribute Changes

(a) The system policy shall specify a set of subject attributes such that any change of those attributes during an interactive session would cause a notification to be sent to the user associated with that subject.

(b) The system policy shall also specify a set of subject attributes that can be displayed to the user as a result of a user's query.

Dependencies:

- Uses: ACA-1
- Policy: ACA-2

ACA-6. Policy Coherence of Access Control Attributes

(a) If different attributes are defined for different subjects and objects, the assignment of these attributes shall be consistent with the defined access control pol-

icy.

Dependencies:

- Uses: ACA-1

ACA-7. Multiple-Policy Support

(a) If multiple access control policies are supported, the access control attributes corresponding to each individual policy shall be identified.

Dependencies:

- Uses: ACA-1

1.2 COMPONENTS

All components of this family include the three key elements of any access control policy, namely the definition and protection of access control attributes, subject attributes, and object attributes. Since some policies allow different attributes to be attached to different subjects and objects, and since some systems support multiple policies, two additional elements, namely the policy coherence of access control attributes and multiple-policy support, are included as conditional requirements of all components. Other components include elements defining additional requirements for attribute definition and change. The three components defined below are rated based on granularity and coverage of the elements in the components.

Component AC_ACA.1. Basic Attribute Definition

This component include the basic elements that are common to all access control policies. It consists of the following elements:

- ACA-1. Definition and Protection of Access Control Attributes
- ACA-2. Subject Attributes
- ACA-3. Object Attributes
- ACA-6. Policy Coherence of Access Control Attributes
- ACA-7. Multiple-Policy Support

Component AC_ACA.2. Notification of Attribute Changes

This component extends AC_ACA.1 by including requirements for user notification and display of current subject attributes and of attribute changes, and separate definition and assignment of physical device attributes to reflect constraints imposed by device location. It consists of the following elements:

- ACA-1. Definition and Protection of Access Control Attributes
- ACA-2. Subject Attributes
- ACA-3. Object Attributes
- ACA-4. Device Attributes
- ACA-5. User Notification of Attribute Changes
- ACA-6. Policy Coherence of Access Control Attributes
- ACA-7. Multiple-Policy Support

Component AC_ACA.2A. Enhanced Attribute Definition

This component enhances AC_ACA.2 by including specific requirements for the assignment of access control attributes to subjects and objects. It consists of the following elements.

- ACA-1. Definition and Protection of Access Control Attributes
- ACA-2. Subject Attributes
- ACA-3A. Enhanced Object Attributes
- ACA-4. Device Attributes
- ACA-5. User Notification of Attribute Changes
- ACA-6. Policy Coherence of Access Control Attributes
- ACA-7. Multiple-Policy Support

It is envisioned that component AC_ACA.1 will be used in most profiles defining access control polices as it includes the common requirements of policy attribute definitions. Component AC_ACA.2 can be used in profiles defining access control policies where user actions may depend on the current sensitivity of the subject's access attributes. For this reason, the requirement for user notification of any change of subject attributes becomes important in such profiles. It is anticipated that component AC_ACA.2A will be used in profiles where the physical environment affects the ranges of sensitivity of physical devices. These environments include those where mandatory access control policies are enforced.

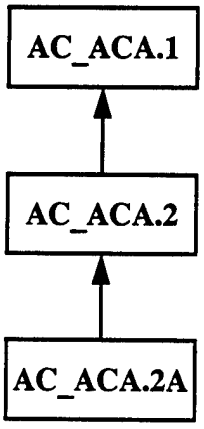


Figure 24. Component Relationships: Access Control Attributes

2. AUTHORIZATION OF SUBJECT ACCESS TO OBJECTS

2.1 ELEMENTS

SAO-1. Specification of Access Authorization Locations

(a) The places where access is checked and granted shall be defined (e.g., reference monitor of a specific host TCB, service, gateway, router).

Dependencies:

- Uses: RM-1

SAO-2. Access Authorization Enforcement

(a) Access authorization shall be enforced by the TCB of each host.
(b) If the system is defined as a set of realms, it shall be possible to use the identity of a subject's realm in the enforcement of access authorization.
(c) If the system security perimeter is defined and enforced in the communication network, access authorization shall also be enforced by each perimeter gateway, router, or network-security device.

Dependencies:

- Uses: SAO-1

SAO-3. Authorization Rules for Access Mediation

(a) The system shall define and enforce authorization rules for the mediation of subject references to objects.
(b) The authorization rules shall be based on the access control attributes and types of subjects and objects.
(c) The authorization rules shall provide protection for objects from unauthorized access, either by explicit user action or by default.
(d) The authorization rules shall specify and enforce controls over the creation and destruction of subject and objects, object reuse, default subject or object attributes and attribute inheritance rules (if any), and resource availability (e.g., storage space shall be available for the creation of a subject and object).

Dependencies:

- Uses: SAO-2

- Policy: ACA-1–ACA-4

SAO-4. Authorization Rules for Object Encapsulation

(a) The system shall define and enforce authorization rules that shall specify and enforce controls over access to encapsulated objects, creation of object subsystems by users, and invocation of encapsulated subsystem whenever encapsulated objects are supported.

(b) The procedure for determining the effective rights to encapsulated objects shall be defined and enforced by each TCB or server, whenever object encapsulation is supported.

Dependencies:

- Uses: SAO-3
- Policy: ACA-1–ACA-4

SAO-5. Authorization Rules for Delegation Chains

(a) The system shall define and enforce authorization rules that shall distinguish whether a subject is the initiator of an action or is an intermediary of a delegation chain whenever delegated access is supported.

(b) The procedure for determining the effective rights of a delegation chain shall be defined and enforced by each TCB or server whenever access-right delegation is supported.

Dependencies:

- Uses: SAO-3
- Policy: ACA-1–ACA-4, IAI-1A, CA-10–CA-12

SAO-6. Scope of Authorization Rules

(a) The authorization rules shall specify the types of subjects, objects (e.g., processes, segments, devices), and associated access control attributes to which they apply.

Dependencies:

- Uses: SAO-3

SAO-6A. Mandated Scope of the Authorization Rules

(a) The authorization rules shall specify the types of subjects, objects (e.g., processes, segments, devices), and associated access control attributes to which they apply.

(b) The authorization rules shall include all subjects, objects (e.g., processes, segments, devices), and associated access control attributes that are directly or

indirectly accessible to subjects external to a TCB.

Dependencies:

- Uses: SAO-3

SAO-6B. Scope of Non-Discretionary Authorization Rules

(a) The authorization rules shall specify the types of subjects, objects (e.g., processes, segments, devices), and associated access control attributes to which they apply.

(b) The authorization rules shall include all subjects, objects (e.g., processes, segments, devices), and associated access control attributes that are directly or indirectly accessible to subjects external to a TCB.

(c) As a consequence, if non-discretionary access control policies are used to control the flow of information between subjects, the scope of the authorization rules shall also include all policy and status attributes of subjects and storage objects (e.g., quotas, object existence, size, access time, creation and modification time, lock status).

Dependencies:

- Uses: SAO-3

SAO-7. Policy Coherence of Authorization Rules

(a) If different rules apply to different types of subjects and objects, the totality of these rules shall be shown to support the defined policy.

Dependencies:

- Uses: SAO-6, SAO-6A, SAO-6B

SAO-8. Multiple-Policy Support

(a) If multiple policies are supported, the authorization rules for each policy shall be defined separately.

(b) Each host TCB shall define and enforce the composition of policies, including the enforcement of the authorization rules (e.g., subject and object type coverage, enforcement precedence).

Dependencies:

- Uses: SAO-6, SAO-6A, SAO-6B

2.2 COMPONENTS

All components defining authorization policies include the following elements: specification of access authorization locations, access authorization enforcement, authorization rules for access mediation, coherence of authorization rules, and multiple-policy support. The latter two elements are conditional in the sense that they apply only in certain systems and environments. Support for object encapsulation and authorization rules for delegation chains are introduced to distinguish components for specific environments. The scope of authorization rules requirements (i.e., subset of all objects, all objects, or all objects and their status attributes) is common to all components, with variants of the base element used to level related components.

The six components defined below are rated based on scope and coverage of the elements for specification and enforcement of authorization rules and the scope of authorization rule application.

Component AC_SAO.1. Basic Authorization for Object Subsets

This component includes all types of authorization requirements in their most basic form. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-6. Scope of the Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules
- SAO-8. Multiple-Policy Support

Component AC_SAO.2. Authorization with Object Encapsulation

This component extends AC_SAO.1 by including authorization rules for object encapsulation. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-4. Authorization Rules for Object Encapsulation
- SAO-6. Scope of the Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules

- SAO-8. Multiple-Policy Support

Component AC_SAO.3. Authorization with Delegation Chains

This component extends AC_SA0.2 by including authorization rules for delegations. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-5. Authorization Rules for Delegation Chains
- SAO-6. Scope of the Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules
- SAO-8. Multiple-Policy Support

Component AC_SAO.4. Authorization with Object Encapsulation and Delegation Chains

This component extends both AC_SA0.2 and AC_SAO.3 by including authorization rules for both object encapsulation and delegations. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-4. Authorization Rules for Object Encapsulation
- SAO-5. Authorization Rules for Delegation Chains
- SAO-6. Scope of the Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules
- SAO-8. Multiple-Policy Support

Component AC_SAO.4A. Mandated Authorization Scope

This component extends AC_SAO.4 by extending the scope of the authorization rules to all objects, not just to a specified subset. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-4. Authorization Rules for Object Encapsulation

- SAO-5. Authorization Rules for Delegation Chains
- SAO-6A. Mandated Scope of the Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules
- SAO-8. Multiple-Policy Support

Component AC_SAO.4B. Non-Discretionary Policies

This component extends AC_SAO.4A by extending the scope of the authorization rules to include all objects and their status attributes. It consists of the following elements:

- SAO-1. Specification of Access Authorization Locations
- SAO-2. Access Authorization Enforcement
- SAO-3. Authorization Rules for Access Mediation
- SAO-4. Authorization Rules for Object Encapsulation
- SAO-5. Authorization Rules for Delegation Chains
- SAO-6B. Scope of Non-Discretionary Authorization Rules
- SAO-7. Policy Coherence of Authorization Rules
- SAO-8. Multiple-Policy Support

It is envisioned that the first four components defined above will be used in profiles for environments where access controls need not cover all subjects and objects, and component AC_SAO.4A will be used whenever access controls must cover all objects. Component AC_SAO.4B will be used in profiles for non-discretionary access control systems where illegal information flow represents a significant threat.

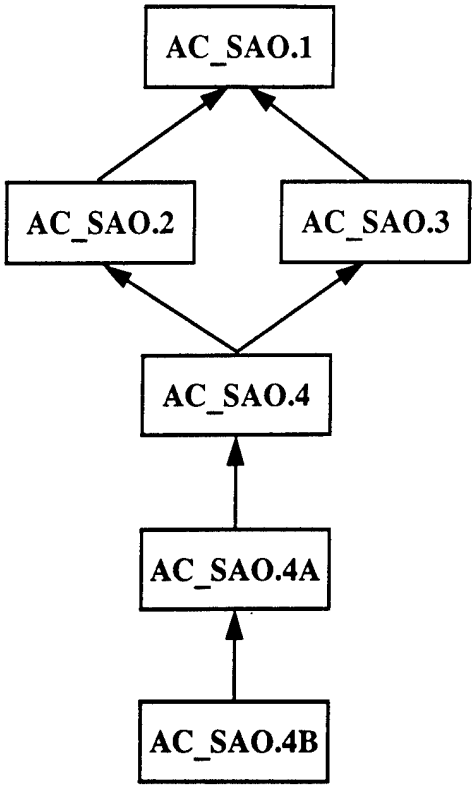


Figure 25. Component Relationships: Subject Access to Objects

3. ADMINISTRATION OF ACCESS CONTROL ATTRIBUTES

3.1 ELEMENTS

AA-1. Assignment and Modification of Attributes

(a) The system shall define and enforce the rules for assignment, modification, and destruction of access control attributes for subjects and objects.

(b) The effect of these rules shall specify that access permission to an object by users not already possessing access permission is assigned only by authorized users.

(c) These rules shall allow authorized users to specify and control sharing of objects by named individuals or defined groups of individuals, or by both, and shall limit propagation of access rights (i.e., these rules shall define the distribution, revocation, and review of access control attributes).

Dependencies:

- Policy: ACA-1–ACA-4

AA-2. Object Import and Export

(a) The rules for assignment, modification, and destruction of access control attributes shall include those for attribute assignment to objects during import and export operations (e.g., import of non-labeled sensitive data, export of labeled information).

Dependencies:

- Policy: ACA-1

AA-3. Support for Delegation Chains

(a) If delegated access is supported, the attribute administration rules shall allow a subject in a delegation chain to delegate a subset of its attributes to another subject.

Dependencies:

- Uses: SAO-3B, SAO-5
- Policy: ACA-1–ACA-4, IAI-1A, CA-10–CA-12

AA-4. Policy Coherence of Attribute Administration Rules

(a) If different rules of assignment, modification, and destruction of access control attributes apply to different subjects, objects, and/or attributes, the totality of these rules shall be shown to support the defined policy.

AA-5. Multiple-Policy Support

(a) If multiple policies are supported, the attribute administration rules for each policy shall be defined separately.

(b) Each host TCB shall define and enforce the composition of policies, including the enforcement of the attribute administration rules (e.g., distribution, review, and revocation of access rights).

3.2 COMPONENTS

The two components defining the administration of access control attributes include the common elements assignment and modification of attributes, object import and export, policy coherence of attribute administration rules, and multiple-policy support. The requirements of the last two elements are conditional in the sense that they apply only in certain systems and environments. The distinguishing element of the two components below is the addition of requirements for delegation chain support, which is also conditional.

Component AC_AA.1. Basic Access Control Administration

This component consists of the basic elements of attribute administration that are common to all profiles:

- AA-1. Assignment and Modification of Attributes
- AA-2. Object Import and Export
- AA-4. Policy Coherence of Attribute Administration Rules
- AA-5. Multiple-Policy Support

Component AC_AA.2. Access Control Administration for Delegation Chains

This component extends AC_AA.1 by including requirements to support delegation chains. It consists of the following elements:

- AA-1. Assignment and Modification of Attributes
- AA-2. Object Import and Export
- AA-3. Support for Delegation Chains
- AA-4. Policy Coherence of Attribute Administration Rules
- AA-5. Multiple-Policy Support

It is envisioned that component AC_AA.1 will be used in most profiles, whereas component AC_AA.2 will be used in profiles for systems using delegation chains in the operational environment.

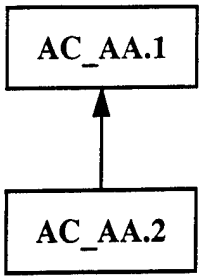


Figure 26. Component Relationships: Administration of AC Attributes

I. COVERT CHANNEL COUNTERMEASURES CLASS

Families:

1. Covert Channel Handling

Covert channel handling includes both technical requirements (e.g., elimination of channels, bandwidth reduction to acceptable levels, deterrence of use by auditing covert storage channels) and administrative or environmental requirements (e.g., exclusive use of trusted software by trusted users in environments where all unauthorized information flow must be prevented).

Covert channel elimination requires that the design and/or implementation of a system be changed so that covert channels are removed from the product. These changes include the elimination of resource sharing between any subjects that could take part in covert channel use by preallocating maximum resource demands to all such subjects or by partitioning resources on a per-subject basis, and the elimination of interfaces, features, and mechanisms which can cause covert leakage of sensitive data. Since covert channel elimination may be impractical for some channels, other handling functions may be useful in a TCB (e.g., bandwidth limitation functions).

Covert channel bandwidth limitation requires that the maximum or, alternatively, the average bandwidth of any channel be reduced to a limit deemed acceptable in the environment of product use. In sensitive applications, bandwidth limitation may require that the aggregated (i.e., combined) bandwidth of a product's covert channels be reduced to an acceptable value. Bandwidths can be limited by (1) deliberate introduction of noise in TCB functions used to exploit the channels (e.g., use of random allocation algorithms for shared resources such as indexes in shared tables, disk areas, and process identifiers, or introduction of extraneous processes that modify covert channel variables of a TCB in pseudo-random patterns), or (2) deliberate introduction of delays in each TCB primitive of a real channel.

Covert channel auditing is a primary method used to discourage the use of covert channels. This method assumes that the frequent use of a channel can be unambiguously detected by audit mechanisms. Some covert channels preclude the use of channel audit, elimination, and bandwidth limitation methods. These channels typically include timing channels that arise from hardware-resource sharing (e.g., shared busses, processor caches). Furthermore, in some environments, threat analysis may indicate that any use of covert channels cannot be tolerated. However, in most commercial products it is impractical to eliminate all covert channels. If such products are used in such non-tolerant environments, the effect of covert channel use must be neutralized. This could be done by the exclusive use of trusted product and application software. In such cases, evidence must be provided to justify that the exclusive use of trusted application software is sufficient to render the existing covert channels ineffective.

1. COVERT CHANNEL HANDLING

1.1 ELEMENTS

CCH-1. Auditability of Covert Channels

(a) The TCB and privileged applications shall include functions that help audit the use of covert storage channels.

(b) These functions shall enable the identification of the transmitter, receiver, and specific covert channels used (e.g., TCB and privileged application element used to transmit information).

Dependencies:

- Uses: LP-1
- Policy: SAO-6B, SPS-3, AE-3

CCH-2. Storage-Channel Audit Functions

(a) The functions added to the TCB and privileged applications for storage-channel auditing shall be identified for each channel and shall be available in common product configurations.

(b) If audit functions are not added to certain storage channels (e.g., hardware storage channels), evidence must be provided to justify why these channels do not represent a security threat for the intended use of the product.

Dependencies:

- Uses: CCH-1
- Policy: SAO-6B, SPS-3, AE-3

CCH-3. Storage-Channel Bandwidth Limitation

(a) TCB functions that help limit the bandwidth and/or eliminate covert storage channels shall be provided.

(b) The bandwidth limits for each channel shall be settable by system administrators.

Dependencies:

- Uses: LP-1, SPS-3, SMT-1

- Policy: SAO-6B, SPS-3

CCH-3A. Storage-Channel Bandwidth Limitation Functions for Common Configurations

(a) TCB and privileged application functions that help limit the bandwidth and/or eliminate covert storage channels shall **also be available in common product configurations.**

(b) The bandwidth limits for each channel shall be settable by system administrators.

(c) **If channel bandwidth limitation and channel elimination functions are not added to certain storage channels (e.g., hardware storage channels), evidence must be provided to justify why these channels do not represent a security threat for the intended use of the product.**

Dependencies:

- Uses: LP-1, SPS-3, SMT-1
- Policy: SAO-6B, SPS-3

CCH-3B. Storage- and Timing-Channel Bandwidth Limitation Functions for Common Configurations

(a) TCB and privileged application functions that help limit the bandwidth and/or eliminate covert storage **or timing** channels shall also be available in common product configurations.

(b) The bandwidth limits for each channel shall be settable by system administrators.

(c) If channel bandwidth limitation and channel elimination functions are not added to certain storage **or timing** channels (e.g., hardware storage channels), evidence must be provided to justify why these channels do not represent a security threat for the intended use of the product.

Dependencies:

- Uses: LP-1, SPS-3, SMT-1
- Policy: SAO-6B, SPS-3

1.2 COMPONENTS

The components of this family include requirements for three types of functions, namely functions for auditing covert channel use, functions for limiting the covert channel bandwidth, and functions for eliminating covert channels. The components provided below illustrate some of the uses of the above functions that would support different covert channel handling policies. The components defined below are rated based on the granularity and coverage of individual elements.

Component CC_CCH.1. Covert Channel Auditing

This component includes the basic requirements for auditing the use of storage channel use. It consists of the following elements:

- CCH-1. Auditability of Covert Channels
- CCH-2. Storage-Channel Audit Functions

Component CC_CCH.2. Covert Channel Auditing and Bandwidth Limitation

This component extends CC_CCH.1 by including a requirement for limiting the bandwidth of (or eliminating) storage channels. It consists of the following elements:

- CCH-1. Auditability of Covert Channels
- CCH-2. Storage-Channel Audit Functions
- CCH-3. Storage-Channel Bandwidth Limitation

Component CC_CCH.2A. Covert Channel Auditing and Bandwidth Limitation for Common Configurations

This component extends CC_CCH.2 by including a requirement for limiting the bandwidth of (or eliminating) storage channels in all common system configurations. It consists of the following elements:

- CCH-1. Auditability of Covert Channels
- CCH-2. Storage-Channel Audit Functions
- CCH-3A. Storage-Channel Bandwidth Limitation Functions for Common Configurations

Component CC_CCH.2B. Covert Storage- and Timing-Channel Auditing and Bandwidth Limitation

This component extends CC_CCH.2A by including a requirement for limiting the bandwidth of (or eliminating) timing channels, not just storage channels, in all common system configurations. It consists of the following elements:

- CCH-1. Auditability of Covert Channels
- CCH-2. Storage-Channel Audit Functions
- CCH-3B. Storage-and Timing-Channel Bandwidth Limitation Functions for Common Configurations

We anticipate that most profiles will use the first two components, CC_CCH.1 and CC_CCH.2, since these components have the least effect on existing applications; i.e., the addition of audit and bandwidth limitation functions to certain TCB configurations may be sufficient to satisfy a given covert channel handling policy. Component CC_CCH.2A can be used in profiles for environments where covert channel handling policies have to be applied uniformly to all system configurations. Component CC_CCH.2B can be used in profiles for environments where leakage of sensitive data via any type of channel is deemed to pose a critical risk.

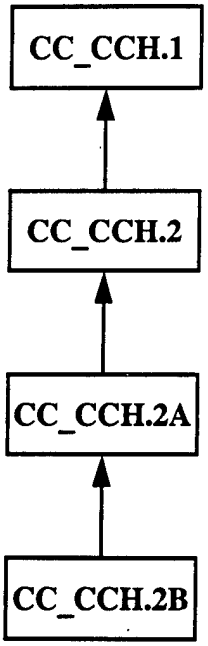


Figure 27. Component Relationships: Covert Channel Handling

J. AUDIT CLASS

Families:

1. **Audit Protection**
2. **Auditable Events**
3. **Audit Capabilities**
4. **Audit Record Structure**
5. **Audit Management**

The audit requirements refer to monitoring and, in some cases, detecting real or potential violations of security policies in organizations that use information technology (IT) products containing audit functions. These functions help monitor the use of access rights by authorized users, and act as a deterrent against usage policy violations.

Auditing involves recognizing, recording, and analyzing user and application actions that are considered, by audit administrators, to be critical to the success of an organization's security policy. The resulting audit records can be examined to determine which security-relevant user actions took place and who was responsible for them. The audit requirements refer to basic audit mechanisms, including audit data protection, record format, event selection, capabilities of the audit function, and audit management, as well as to analysis tools, violation alarms, and real-time intrusion detection systems, which use the basic mechanisms. Recognition of auditable actions is based largely on administratively supplied specifications of user actions and patterns of behavior whose appropriateness is considered to be significant to the satisfaction of an organization's security policy. The designers of an IT product must either anticipate which actions and patterns are likely to be considered important to organizations with respect to their security policies, or provide an audit interface that allows trusted (and possibly other) applications to record and protect audit data, and to perform intrusion detection. Application-provided audit requires separate logging functions and audit trails to prevent applications from interfering with the base audit functions.

Since the purpose of the audit mechanism is to audit user and application actions, including administrative actions, designers of the audit mechanism cannot uniformly assume that all authorized actions are appropriate; consequently, some administrative actions must always be audited.

The IT product must record each action that has been deemed auditable along with accompanying information needed to understand the apparent purpose or effect of that action (e.g., user environment variables, programs used to pre-process user input). Recorded audit data must be protected by each host TCB and audit server from inappropriate modification, use, or destruction. Furthermore, the confidentiality and integrity of the audit data must be guaranteed during transfer to and from audit servers. To avoid repudiation, the mechanism by which audit data is gathered, stored, and processed must be publicly known and reliable. Often this implies the use of secure communication channels. At higher levels of functionality, the auditing of key administrative actions should resist all attacks by online users and otherwise undetectable attacks by users with access to the physical audit media (e.g., through the use of write-once audit disks).

Finally, audit data must be available for analysis in a timely manner and in a useful format within policy constraints established for the product. This requirement motivates the design of pre- and post-processing software that organizes audit data into a presentable format and/or delivers it to authorized users or processes acting on their behalf.

1. AUDIT PROTECTION

1.1 ELEMENTS

AP-1. Authenticated Subject Identity

(a) The identity of the subject generating audit records shall be authenticated before records are appended to the audit trail.

Dependencies:

- Uses: IAI-1–IAI-4, UA-1, IRA-1

AP-2. Audit Trail Protection

(a) The TCB of each host of the distributed system shall be able to create, maintain, and protect from modification, unauthorized access, or destruction an audit trail of accesses to the objects it protects.

Dependencies:

- Uses: LP-1, PP-2–PP-2B

AP-3. Audit Data Confidentiality

(a) The audit data shall be protected by each host TCB so that read access to it is limited to those who are authorized for audit data.

Dependencies:

- Uses: LP-1, PP-2–PP-2B

AP-4. Separation of Trusted Application Audit

(a) If the audit facility (its selection function, logging facility, etc.) is made available to the trusted applications, the trusted application audit mechanism shall use a separate audit logging service and a separate audit trail to prevent applications from interfering (e.g., denying service by filling the audit trail) with the base-system TCB audit.

AP-5. Protected Audit Data Transmission

(a) The confidentiality and integrity of the audit data shall be maintained while

they are transmitted among the TCBs of a distributed system.

Dependencies:

Uses: DCF-1–DCF-6, DIF-1–DIF-6

AP-6. Audit Server Separation

(a) In a distributed system, a separate audit server shall be available and protected from interference and tampering by unauthorized subjects.

(b) The audit server shall be capable of supporting the audit trail management and analysis tools.

(c) System administrators shall be able to specify the event records that are sent to the audit server.

Dependencies:

- Uses: SI-1, SI-2, MPA-6

1.2 COMPONENTS

All audit protection components include a core of four basic elements that are necessary regardless of whether audit is used in a centralized or a distributed system. These elements are authenticated subject identity, audit trail protection, audit data confidentiality, and separation of trusted application audit. The additional two elements refer to audit protection in distributed systems. The three audit protection components defined below are rated based on coverage of the elements in the components.

Component AU_AP.1. Basic Audit Protection

This component is intended to cover the basic elements of audit trail and audit data protection that are necessary for all profiles. This component consists of the following elements:

- AP-1. Authenticated Subject Identity
- AP-2. Audit Trail Protection
- AP-3. Audit Data Confidentiality
- AP-4. Separation of Trusted Application Audit

Component AU_AP.2. Separation of Application Audit

This component includes all the elements of AU_AP.1 and, in addition, requires that the audit data be protected during transmission among TCBs. This component consists of the following elements:

- AP-1. Authenticated Subject Identity
- AP-2. Audit Trail Protection
- AP-3. Audit Data Confidentiality
- AP-4. Separation of Trusted Application Audit
- AP-5. Protected Audit Data Transmission

Component AU_AP.3. Separation of Audit Servers

This component consists of all the following elements of audit protection:

- AP-1. Authenticated Subject Identity
- AP-2. Audit Trail Protection
- AP-3. Audit Data Confidentiality
- AP-4. Separation of Trusted Application Audit
- AP-5. Protected Audit Data Transmission

- AP-6. Audit Server Separation

It is envisioned that component AU_AP.1 will be used in the majority of centralized system profiles, whereas components AU_AP.2 and AU_AP.3 will be used for distributed systems profiles.

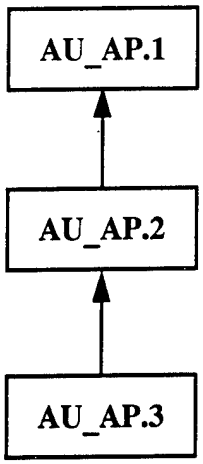


Figure 28. Component Relationships: Audit Protection

2. AUDITABLE EVENTS

2.1 ELEMENTS

AE-1. Access Control and Administrative Policy Events

(a) The TCB of each host of the distributed system shall be able to record the following types of access control and administrative policy events:

1. **Use of the identification and authentication function and system entry events;**
2. **Access control events selectable on a per-user, per-subject, per-object, per-realm, and per-policy attribute basis; i.e., introduction of objects into a user's address space (e.g., file open, program initiation), creation and deletion of subjects and objects; distribution and revocation of access rights; changes of user, subject, object, and realm policy attributes; acquisition and deletion of TCB privileges; and**
3. **Administrative events caused by actions taken by computer operators and system administrators and/or system security officers; e.g., privileged operations such as the modification of TCB elements; accesses to TCB objects; changes of policy attributes of users, TCB configuration and security characteristics, and system privileges; changes of the realm configuration caused by hosts leaving and rejoining a network; selection and modification of audited events.**

Dependencies:

- Uses: IAI-1–IAI-4, UA-1, IRA-1, DSE-2, DSE-3, ACA-1–ACA-4, SOA-1–SOA-5, AA-1–AA-4, SPS-1–SPS-7, MPA-1–MPA-8

AE-2. Data Confidentiality and Integrity Policy Events

(a) The TCB of each host of the distributed system shall be able to record the following types of **data confidentiality and integrity** policy events:

1. **Communication events (e.g., establishing a connection or a connectionless association between two hosts, or terminating a connection); and**
2. **Data confidentiality and integrity events (e.g., detection of confidentiality or integrity violations including message modifications or replays).**

Dependencies:

- Uses: AE-1, IAI-1–IAI-4, UA-1, IRA-1, DSE-2, DSE-3, ACA-1–ACA-4, SOA-1–SOA-5, AA-1–AA-4, DCF-1–DCF-6, DIF-1–DIF-6

AE-3. Non-Discretionary Policy Events

(a) If non-discretionary access control policies are supported, the TCB of each host shall be able to record any override of human-readable output markings.

(b) When the non-discretionary access control policies to control the flow of information between subjects are supported, the TCB of each host shall also be able to audit the identified events that may be used in the exploitation of covert storage channels.

Dependencies:

- Uses: AE-1, ACA-1–ACA-4, CCH-1–CCH-3

AE-4. Availability Policy Events

(a) If availability policies are supported, the TCB of each host of the distributed system shall be able to record attempts to circumvent or otherwise gain unauthorized access to resource-allocation limits.

Dependencies:

- Uses: AE-1, Availability[16]

AE-5. Cryptographic Policy Events

(a) If cryptographic policies are supported, the TCB of each host of the distributed system shall be able to record the following types of cryptographic policy events:

1. **Key-management events (e.g., generation, distribution, installation, use, and maintenance of cryptographic keys);**
2. **Privileged cryptographic operations executed by system administrators such as the modification of cryptographic-domain elements; and**
3. **Use of cryptographic functions (e.g., data encryption, decryption with an incorrect key).**

Dependencies:

- Uses: AE-1, CDP-1–CDP-2, SKM-1–SKM-11

[16] Pending development of availability requirements.

AE-6. Default and Dependent Events

(a) The following events shall be defined:

1. **Events that are auditable by default, and**
2. **Events that are required for successful auditing of other events that may not be disabled.**

Dependencies:

- Uses: AE-1

2.2 COMPONENTS

All components defining the auditable event sets include two common elements, namely one defined for access control and administrative policy events and another defined for default and dependent events. The rest of the components are rated based on the coverage of auditable event elements. For example, some elements include distributed system requirements whereas other components include policy-specific requirements. The components provided below illustrate a representative group of auditable event components.

Component AU_AE.1. Basic Auditable-Event Set

This component is intended to cover the minimum set of auditable-event requirements. It consists of the following elements:

- AE-1. Access Control and Administrative Policy Events
- AE-6. Default and Dependent Events

Component AU_AE.2. Extended Auditable-Event Set

This component extends AU_AE.1 by adding the non-discretionary and availability policy events to the set of auditable events that must be supported in any system. It consists of the following elements:

- AE-1. Access Control and Administrative Policy Events
- AE-3. Non-Discretionary Policy Events
- AE-4. Availability Policy Events
- AE-6. Default and Dependent Events

Component AU_AE.3. Auditable-Event Set for Distributed Systems

This component extends AU_AE.1 by adding non-discretionary policy events and requiring that data confidentiality and integrity events be covered in distributed systems. This component consists of the following elements:

- AE-1. Access Control and Administrative Policy Events
- AE-2. Data Confidentiality and Integrity Policy Events
- AE-3. Non-Discretionary Policy Events
- AE-6. Default and Dependent Events

Component AU_AE.4. Extended Auditable-Event Set for Distributed Systems

This component extends both AU_AE.2 and AU_AE.3 by adding the requirement that cryptographic policy events be covered, on an as-needed basis, in distributed systems. This component consists of the following elements:

- AE-1. Access Control and Administrative Policy Events
- AE-2. Data Confidentiality and Integrity Policy Events
- AE-3. Non-Discretionary Policy Events
- AE-4. Availability Policy Events
- AE-5. Cryptographic Policy Events
- AE-6. Default and Dependent Events

It is envisioned that component AU_AE.1 will be used in the majority of profiles that require basic auditing for distributed systems. This component can be instantiated in profiles supporting default event auditing for different access control policies. Component AU_AE.2 extends the coverage of the requirements of AU_AE.1 by enlarging the set of events to include non-discretionary and availability policy events. This component can be used in profiles for transaction systems. Component AU_AE.3 is intended for use in distributed systems where confidentiality and integrity events may also become auditable. Finally, AU_AE.4 is intended for use in systems where the audit of cryptographic policy events may be necessary to detect specific operational aspects of cryptographic policy enforcement and use.

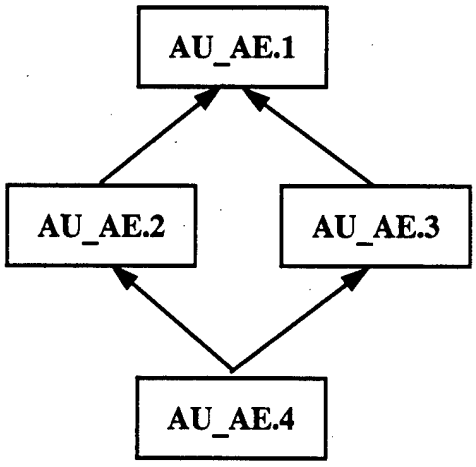


Figure 29. Component Relationships: Auditable Events

3. AUDIT CAPABILITIES

3.1 ELEMENTS

AC-1. Event Selection Display

(a) The TCB of each host shall provide a protected function that displays the currently selected events and their defaults.

(b) The use of this function shall be restricted to authorized system administrators.

Dependencies:

- Uses: AE-1

AC-2. Event Class Modification

(a) System administrators shall be able to modify existing event classes and/or the introduce new event classes without generating a new TCB for one or all hosts of the distributed system (e.g., recompilation of TCB elements shall not be required).

Dependencies:

- Uses: AE-1, AC-1

AC-3. Per-Subject and Per-Object Event Selection

(a) The audit function shall be capable of supporting per-subject and/or per-object event selection.

Dependencies:

- Uses: AE-1, AC-1

AC-4. Event Exclusion Auditing

(a) Event exclusion controls shall be supported (e.g., audit all subjects of realm X except group Y in X).

Dependencies:

- Uses: AE-1, AC-1

AC-5. Event Accumulation and Handling

(a) The TCB of each host shall contain a function able to monitor the occurrence or accumulation of locally auditable events that may indicate an attempted violation of the product's security policy.

(b) The TCB of each host shall be able to notify the security administrator when specified accumulation thresholds are exceeded.

(c) If the occurrence or accumulation of these security-relevant events continues, the system shall take the least disruptive action to terminate the event (i.e., as a default action, the TCB of each host shall be able to send an alarm message to the system console and/or the administrator's terminal when specified thresholds are exceeded).

(d) System administrators shall be able to specify the action to be taken when accumulation thresholds are exceeded and audit trail records cannot be recorded (e.g., initiate a secure shutdown of the subject generating those events, stop the logging service, discard or overwrite old audit trails, shut down the host).

Dependencies:

- Uses: AE-1, AC-1

AC-5A. Event Accumulation and Immediate Handling

(a) The TCB of each host shall contain a function able to monitor the occurrence or accumulation of locally auditable events that may indicate an attempted violation of the product's security policy.

(b) The TCB of each host shall **immediately** notify the security administrator when specified accumulation thresholds are exceeded.

(c) If the occurrence or accumulation of these security-relevant events continues, the system shall take the least disruptive action to terminate the event (i.e., as a default action, the TCB of each host shall be able to send an alarm message to the system console and/or the administrator's terminal when specified thresholds are exceeded).

(d) System administrators shall be able to specify the action to be taken **immediately after** accumulation thresholds are exceeded and audit trail records cannot be recorded (e.g., initiate a secure shutdown of the subject generating those events, stop the logging service, discard or overwrite old audit trails, shut down the host).

Dependencies:

- Uses: AE-1, AC-1

AC-6. Real-Time Event Reporting and Intrusion Detection

(a) The TCB of each host shall also be able to perform real-time event reporting

and intrusion detection in support of the product's security policy.

Dependencies:

- Uses: AE-1, AC-1, AC-5A

3.2 COMPONENTS

All components defining the auditing function include three basic elements, namely event selection display, event class modification, and per-subject and per-object event selection. The first two elements enable the basic operation of all auditing functions whereas the third requires the capability of selecting events related to a specific subject, a specific object, or both. The rest of the components include elements that extend auditable event selection and monitoring. The components of this function are rated based on the coverage of auditable-event elements.

Component AU_AC.1. Basic Audit Function

This component includes the basic, common requirements for all audit functions. It consists of the following elements:

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection

Component AU_AC.2. Extended Audit Function

This component consists of all the basic audit elements of AU_AC.1, and extends AU_AC.1 by including an element intended to ease the specification of auditable events.

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection
- AC-4. Event Exclusion Auditing

Component AU_AC.3. Event Accumulation Auditing

This component extends AU_AC.1 by including requirements for the detection and handling of event accumulations. In particular, it levies requirements for event accumulation handling on both the TCB and system administrator functions. This component consists of the following elements:

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection
- AC-5. Event Accumulation and Handling

Component AU_AC.3A. Immediate Event-Accumulation Auditing

This component extends AU_AC.3 only by requiring immediacy of the system's event-accumulation notification and response capabilities. This component consists of the following elements:

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection
- AC-5A. Event Accumulation and Immediate Handling

Component AU_AC.4. Real-Time Auditing and Intrusion Detection

This component extends AU_AC.3A in that it requires a capability for real-time event reporting and for intrusion detection. It consists of the following elements:

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection
- AC-5A. Event Accumulation and Immediate Handling
- AC-6. Real-Time Event Reporting and Intrusion Detection

Component AU_AC.5. Extended Auditing in Real Time

This component represents the union of both AU_AC.2 and AU_AC.4 elements, and it represents the most stringent set of audit requirements based on the given elements. This component consists of the following elements:

- AC-1. Event Selection Display
- AC-2. Event Class Modification
- AC-3. Per-Subject and Per-Object Event Selection
- AC-4. Event Exclusion Auditing
- AC-5A. Event Accumulation and Immediate Handling
- AC-6. Real-Time Event Reporting and Intrusion Detection

It is envisioned that components AU_AC.1 and AU_AC.2 will be used in the majority of the profiles whose environment of use does not require event accumulation auditing. Components AU_AC.3 through AU_AC.4 can be used in profiles whose environment of use requires different degrees of real-time response to both individual events and to event accumulations, as well as intrusion detection. It is anticipated that component AU_AC.5

will be used in the same environments where flexible event specification, real-time event notification and response, and intrusion detection are needed.

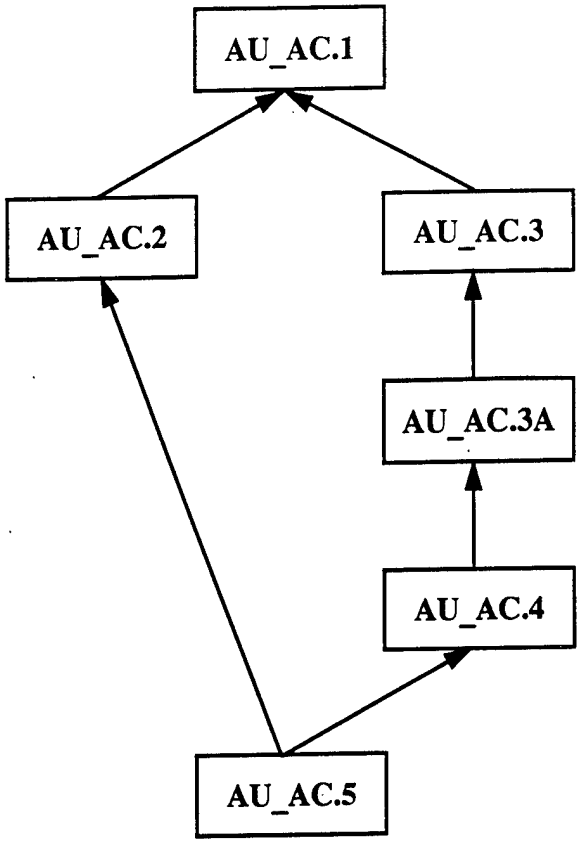


Figure 30. Component Relationships: Audit Capabilities

4. AUDIT RECORD STRUCTURE

4.1 ELEMENTS

ARS-1. Common Audit Record Data

(a) For each recorded event, the audit record shall identify:

1. **Date and time of the event,**
2. **Subject attribute information (e.g., subject's identifier, security or integrity levels, groups, realm, and delegation chain),**
3. **Identity of the host TCB generating the audit record (e.g., the registration, authentication, or time server),**
4. **Event class and event identifier within the class, and**
5. **Event outcome (e.g., success or failure of an action).**

Dependencies:

- Uses: AE-1

ARS-2. Audit Record Data for Object Creation and Destruction

(a) For events that introduce an object into a user's address space and for object deletion events, the audit record shall include the name and policy attributes of the object (e.g., object security level).

Dependencies:

- Uses: AE-1, ARS-1

ARS-3. Authentication Audit Record Data

(a) For identification and authentication and system-entry events, the audit record shall include the following:

1. **The subject's authentication status (e.g, unauthenticated, authenticated with a name, or authenticated with a privilege certificate), and**
2. **The subject's system entry attributes (e.g., console, tty, dial-in, network address).**

Dependencies:

- Uses: AE-1, ARS-1

ARS-3A. Audit Record Data for Subject Aliasing

(a) For identification and authentication and system-entry events, the audit record shall include the following:

1. The subject's authentication status (e.g, unauthenticated, authenticated with a name, or authenticated with a privilege certificate), and
2. The subject's system entry attributes (e.g., console, tty, dial-in, network address).

(b) If the subject anonymity is required, the audit record shall include an alias for the subject identity and its location (e.g., login host).

Dependencies:

- Uses: AE-1, ARS-1

4.2 COMPONENTS

There are two components defining the requirements of the audit record structure. The first component includes the basic records structure, whereas the second component extends the first one by including aliasing requirements for the environments where anonymity is desired.

Component AU_ARS.1. Basic Record Structure

This components includes the common requirements for all audit record structures. It consists of the following elements:

- ARS-1. Common Audit Record Data
- ARS-2. Audit Record Data for Object Creation and Destruction
- ARS-3. Authentication Audit Record Data

Component AU_ARS.1A. Extended Record Structure

This component extends AU_ARS.1 by including a requirement for aliasing subject identities. This component consists of the following elements:

- ARS-1. Common Audit Record Data
- ARS-2. Audit Record Data for Object Creation and Destruction
- ARS-3A. Audit Record Data for Subject Aliasing

It is envisioned that most profiles will use component AU_ARS.1. Use of component AU_ARS.1A is anticipated in profiles that include subject anonymity requirements.

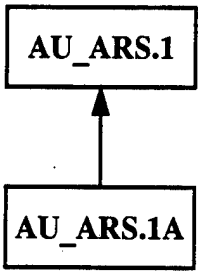


Figure 31. Component Relationships: Audit Record Structure

5. AUDIT MANAGEMENT

5.1 ELEMENTS

AM-1. Basic Audit Management

(a) During the distributed system operation, the TCB of each host shall provide a protected function to turn auditing on and off, and to select and change the events to be audited and their defaults.

(b) Use of this function shall be restricted to authorized system administrators.

Dependencies:

- Uses: AE-1, LP-1, PP-2–PP-2B, RM-3, SAR-1

AM-1A. Basic Audit Management with Single Login

(a) During the distributed system operation, the TCB of each host shall provide a protected function to turn auditing on and off, and to select and change the events to be audited and their defaults.

(b) Use of this function shall be restricted to authorized system administrators.

(c) Use of this function shall not require a separate login to each host of the distributed system.

Dependencies:

- Uses: AE-1, LP-1, PP-2–PP-2B, RM-3, SAR-1, UA-9

AM-2. Basic Audit Selection Functions

(a) Authorized system administrators shall be able to perform the following tasks:

1. **Define auditable-event classes,**
2. **Perform audit selection based on host identity and subject policy attributes (e.g., subject identity, security level), and**
3. **Perform audit selection based on specified accesses on individual objects and object policy attributes (e.g., object security level).**

Dependencies:

- Uses: AM-1, AM-1A

AM-2A. Enhanced Audit Selection Functions

(a) Authorized system administrators shall be able to perform the following tasks:

1. Define auditable-event classes,
2. Perform audit selection based on host identity and subject policy attributes (e.g., subject identity, security level),
3. Perform audit selection based on specified accesses on individual objects and object policy attributes (e.g., object security level),
4. **Perform audit selection based on the outcome of events (e.g., audit records shall be generated only when events succeed or when events fail because of access denial), and**
5. **Specify additional actions to be taken in addition to the generation of an audit record (e.g., to specify that some event records shall be sent to the their console, to the audit trail, or both).**

Dependencies:

- Uses: AM-1, AM-1A, ARS-1

AM-3. Audit Trail Management

(a) Tools for audit trail management shall be provided for use by authorized system administrators.

(b) These tools shall enable the following management functions to be performed:

1. **Creation, destruction, and emptying of audit trails;**
2. **Use of warning points regarding the size of the audit data, and modification of the audit trail size;**
3. **Display of formatted audit trail data;**
4. **Selection of hosts intended to act as audit servers, whenever separate audit servers are supported; and**
5. **Maintenance of audit-trail consistency in the face of TCB (or audit server) failures and discontinuity of operation.**

Dependencies:

- Uses: AM-1, AM-1A

AM-4. Audit Data Management

(a) The TCB of each host shall provide tools for audit data management.

(b) These tools shall be capable of the following:

1. **Verifying the consistency of the audit data,**
2. **Verifying the selection of audit events, and**
3. **Formatting and compressing of event records.**

Dependencies:

- Uses: AM-1, AM-1A, AM-3

AM-5. Audit Review Tools with Intrusion Detection

(a) Audit review tools shall be available to authorized system administrators to assist in the inspection and review of audit data, and shall be protected from unauthorized use, modification, or destruction.

(b) Tools shall also be provided for post-collection audit analysis (e.g., intrusion detection), and shall be able to selectively review the following:

1. **Actions of one or more subjects (e.g., identification, authentication, system entry, and access control actions) on a specified set of hosts;**
2. **Actions performed on a specific object or system resource, whenever object auditing is supported;**
3. **Actions associated with a specific policy attribute, whenever policy-attribute auditing is supported;**
4. **Events with some specific action outcome(s);**
5. **Events that occurred during a specified time period; and**
6. **Events with specified event identifier.**

Dependencies:

- Uses: AM-1, AM-1A, AM -4, AE-1–AE-6, ARS-1–ARS-3, SMT-1, LP-1, PP-2–PP-2B

AM-6. Audit Review Support in Distributed Systems

(a) The review tools shall be able to operate concurrently with the distributed system operation.

Dependencies:

- Uses: AM-1, AM-1A, AM-5

5.2 COMPONENTS

Most audit management components include three common elements, namely basic audit management, basic audit selection functions, and audit trail management. Several components include extensions to these basic elements, such as those of multi-TCB audit management with a single login, and extended auditable-event selection functions. Several components include elements of audit data management and audit review tools. These elements are necessary in all but the most rudimentary audit systems where audit data reduction and review are performed using generic editors rather than specific management tools. The audit management components are rated based on the scope and coverage of individual elements.

Component AU_AM.1. Minimal Audit Management

This component includes the minimal requirements necessary to audit events in a computing system. It includes requirements for turning on and off audit, for event selection capability, and for defining the scope of audit events (i.e., per-subject or per-object audit). This component consists of the following elements.

- AM-1. Basic Audit Management
- AM-2. Basic Audit Selection Functions

Component AU_AM.2. Basic Audit Management

This component extends AU_AM.1 by including audit trail management requirements. This component consists of the following elements:

- AM-1. Basic Audit Management
- AM-2. Basic Audit Selection Functions
- AM-3. Audit Trail Management

Component AU_AM.2A. Basic Audit Management with Single Login

This component extends AU_AM.1 by including a requirement for single login for all hosts of the distributed system. This component consists of the following elements:

- AM-1A. Basic Audit Management with Single Login
- AM-2. Basic Audit Selection Functions
- AM-3. Audit Trail Management

Component AU_AM.3. Audit Data Management

This component extends AU_AM.1 by including an element for audit data (as opposed to audit trail) management. While audit data management may be possible with generic editors, specialized tools significantly enhance the auditors' capability to manage audit data. This component consists of the following elements:

- AM-1. Basic Audit Management
- AM-2. Basic Audit Selection Functions
- AM-3. Audit Trail Management
- AM-4. Audit Data Management

Component AU_AM.4. Audit Data Management with Review Tools

This component extends AU_AM.3 by including an element for audit review and intrusion detection. While audit review and intrusion detection may be possible with generic editors, specialized tools significantly enhance the auditors' capability to audit unusual events and event accumulations, and to detect intrusions. This component consists of the following elements:

- AM-1. Basic Audit Management
- AM-2. Basic Audit Selection Functions
- AM-3. Audit Trail Management
- AM-4. Audit Data Management
- AM-5. Audit Review Tools with Intrusion Detection

Component AU_AM.4A. Enhanced Audit Data Management with Review Tools

This component extends AU_AM.4 by extending (1) the audit selection requirements to include selection based on event outcomes and to include additional actions to be taken automatically for the generation of an audit record (i.e., audit record redirection), and (2) the basic audit management function to include single login in a distributed system. This component consists of the following elements:

- AM-1A. Basic Audit Management with Single Login
- AM-2A. Enhanced Audit Selection Functions
- AM-3. Audit Trail Management
- AM-4. Audit Data Management
- AM-5. Audit Review Tools with Intrusion Detection

Component AU_AM.5. Audit Data Management in Distributed Systems

This component extends AU_AM.4A by requiring that the audit review tools operate concurrently with the distributed system operation. It consists of the following elements:

- AM-1A. Basic Audit Management with Single Login
- AM-2A. Enhanced Audit Selection Functions
- AM-3. Audit Trail Management
- AM-4. Audit Data Management
- AM-5. Audit Review Tools with Intrusion Detection
- AM-6. Audit Review Support in Distributed Systems

It is envisioned that the audit management components AU_AM.1, AU_AM.2, and AU_AM.3 will be used in most profiles. AU_AM.2A is the most basic component for distributed system audit. Whenever specialized audit data review and intrusion detection are deemed necessary, components AU_AM.4, AU_AM.4A, and AU_AM.5 are desirable.

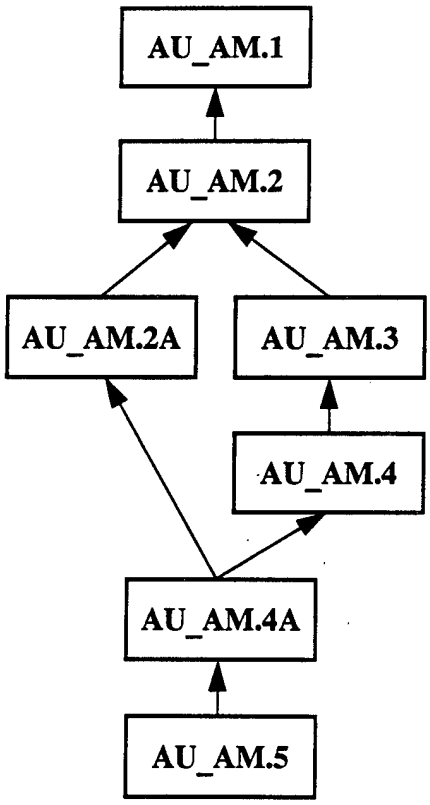


Figure 32. Component Relationships: Audit Management

K. AVAILABILITY CLASS

TBD.

Due to a weaker technical foundation for this class of functional requirements, no requirements for Availability were attempted for this study. The inclusion of this (empty) class simply acknowledges this as an increasingly important area.

L. SECURITY MANAGEMENT CLASS

Families:

1. **Secure Installation**
2. **Security Policy Selection**
3. **Management of Policy Attributes**
4. **Separation of Administrative Roles**
5. **Security Management Tools**

An information technology product must support security management functions to enable administrative users to set up and control the secure operation of the product. The requirements provided in this area refer to TCB functions associated with both administrator and operator roles, and have direct relevance for the security policy selection and enforcement in a system.

Security management components refer to the following types of functions:

a. Host TCB generation, installation, configuration, and maintenance (e.g., recovery, repair of damaged TCB hardware and software elements).

b. Policy selection and update of security policy parameters (e.g., identification and authentication, system entry, access control, cryptographic function, data confidentiality and integrity, and availability parameters).

c. Definition and update of user security characteristics (e.g., unique identifiers associated with user names, user accounts, per-user policy attributes, system entry parameters, availability parameters, or resource quotas).

d. Routine control and maintenance of product resources (e.g., enable and disable peripheral devices, mounting of removable storage media, backup and recovery of user objects, and routine maintenance of host TCB hardware and software elements).

e. Auditing both privileged and unprivileged user actions, and audit management (e.g., selection of audit events, management of audit trails, audit trail analysis, and audit report generation).

Security management functions help counter the same threats as those countered by the security policy functions, including identification and authentication, system entry, trusted path, access control, secure host TCB interconnection, cryptographic support, data confidentiality and integrity, and availability. This is the case because the security management functions implement a significant part of all the system security policies. In addition, when the security management functions are partitioned into different administrative roles, they help limit the potential damage caused by unskilled or malicious administrators.

1. SECURE INSTALLATION

1.1 ELEMENTS

SI-1. Installation and Start-Up Functions

(a) Each host TCB shall perform the following:

1. **Provide an installation function for setting and updating TCB configuration parameters, internal databases, and tables;**
2. **Distinguish between normal and maintenance modes of operation; and**
3. **Start up and recover the system in the maintenance mode.**

Dependencies:

- Uses: LP-1, EU-1

SI-2. Basic Configuration Functions

(a) Administrative functions shall include functions to perform the following:

1. **Initialize, display, modify, and delete configuration parameters;**
2. **Initialize protection-relevant data structures;**
3. **Configure administrative databases; and**
4. **Establish a minimal system configuration before any user or administrator policy attributes are initialized.**

Dependencies:

- Uses: SI-1

SI-2A. Enhanced Configuration Functions

(a) Administrative functions shall include functions to perform the following:

1. Initialize, display, modify, and delete configuration parameters;
2. Initialize protection-relevant data structures;
3. Configure administrative databases; and
4. Establish a minimal system configuration before any user or administrator policy attributes are initialized.

(b) Administrative functions shall also allow the initialization of the following:

1. **Identification and authentication attributes for system administrators;**
2. **System-entry attributes for system administrators; and**

3. **Privileges for separate administrative roles, whenever such roles are provided.**

Dependencies:

- Uses: SI-1, DSE-3, PO-1A

1.2 COMPONENTS

The secure installation components include all the basic requirements for system installation, start up, and initial configuration. The two components are distinguished by the requirement for initialization of special attributes for administrator roles and privileges. The secure installation components defined below are rated based on coverage of the elements in the components.

Component SM_SI.1. Basic Installation and Configuration

This component is intended to cover the basic elements of system installation that are necessary for all profiles. This component consists of the following elements:

- SI-1. Installation and Start-Up Functions
- SI-2. Basic Configuration Functions

Component SM_SI.1A. Enhanced Installation and Configuration

This component includes all the requirements of SM_SI.1 and, in addition, requires a system installation function capable of initializing separate data structures for different administrative roles. This component consists of the following elements:

- SI-1. Installation and Start-Up Functions
- SI-2A. Enhanced Configuration Functions

It is envisioned that component SM_SI.1 will be used in the majority of centralized system profiles, whereas SM_SI.1A will be used only when separate administrative roles are supported.

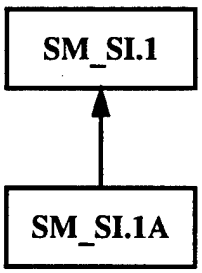


Figure 33. Component Relationships: Secure Installation

2. SECURE POLICY SELECTION

2.1 ELEMENTS

SPS-1. Scope of Security Policy Management

(a) Administrative functions shall be provided by the TCB of a centralized-system product and by each realm of a distributed-system product if distributed systems are partitioned into separate administrative realms.

(b) These functions shall enable system administrators to select system policies (e.g., by selecting the appropriate policy attributes).

(c) The use of these functions shall be restricted to authorized administrative personnel.

Dependencies:

- Uses: LP-1

SPS-2. Identification and Authentication Policy Selection

(a) Administrative functions shall enable system administrators to select identification and authentication policies.

(b) These functions shall include selection of the following:

1. **The login authentication method (e.g., passwords, tokens, or biometrics) and policy options (e.g., whether certified login is required) on a per policy-attribute basis whenever multiple identification and authentication methods can be used.**
2. **The authenticated boot image of the host TCB (e.g., local or network boot),**
3. **The allowed subject authentication policy options (e.g., for certificate or ticket delegation, postdating, revocation, and renewal),**
4. **The user registration policy options (e.g., single- or multiple-realm registration),**
5. **Privacy policy options (e.g., whether anonymous but traceable user operation shall be allowed or supported), and**
6. **The intra- and inter-realm channel authentication policy.**

Dependencies:

- Uses: IAP-1A
- Policy: SPS-1

SPS-3. Access Control Policy Selection

(a) Administrative functions shall enable system administrators to select access control policies for the current system configuration (e.g., discretionary access control, role-based access control, mandatory access control, router-based access control).

Dependencies:

- Uses: SAO-3
- Policy: SPS-1

SPS-4. TCB Interconnection Policy Selection

(a) Administrative functions shall enable system administrators to select the interconnection policies for the TCBs of the distributed system.

(b) These functions shall include the following:

1. **Selection of the security-perimeter enforcement policy and policy options of the realm gateways,**
2. **Selection of the integrity and sensitivity policies for the information stored within each host TCB,**
3. **Selection of an authentication path policy (e.g., a policy defining valid inter-realm authentication paths),**
4. **Selection of a configuration policy for the realm's security-relevant servers (e.g., registration, authorization, authentication, audit, time, and directory servers), and**
5. **Selection of an availability policy for the realm's security-relevant servers.**

Dependencies:

- Uses: IAP-1A, SAO-3, SI-2
- Policy: SPS-1, SPS-2, SPS-3, DIP-1

SPS-5. Audit Policy Selection

(a) Administrative functions shall enable security administrators to select audit policies (e.g., options denoting per-subject or per-object auditing, auditable security events, event-accumulation thresholds, audit trail thresholds denoting audit-trail warning points and backup intervals).

Dependencies:

- Uses: AE-5
- Policy: SPS-1

SPS-6. Availability Policy Selection

(a) Administrative functions shall enable security administrators to select availability policies (e.g., options denoting choices of trusted recovery, choices of resource allocation and fault tolerance).

Dependencies:

- Policy: SPS-1, Availability[17]

SPS-7. Cryptographic Policy Selection

(a) Administrative functions shall enable system administrators to select cryptographic policies for the following:

1. **Key management functions (e.g., privileges for master key installation or loading, key change, establishing key accounts, key lifetime extensions, key storage and key protection, auditing the key use);**
2. **Selection of cryptographic algorithms, and checksum or signature functions; and**
3. **Data confidentiality use.**

Dependencies:

- Uses: SCF-1, SCF-5
- Policy: SPS-1, DCP-1

[17] Pending development of availability requirements.

2.2 COMPONENTS

The components of this family refer to systems implementing multiple policies of the same type, thereby requiring system administrators to select the policies desired for a particular system configuration. All components include the selection requirements for the basic security policies, including identification and authentication, access control, audit, and interconnection policies for distributed systems. Other components adding requirements for availability and cryptographic policy selection are included. However, other components are possible for cases when only some policy types offer a selection choice whereas others do not. Also, we note that this family is relevant only when a choice of policies of the same type is offered as is the case with many access control and identification and authentication policies, for instance. The components presented below are rated based on the coverage of individual policy selection elements.

Component SM_SPS.1. Selection of Basic Policies

This component is intended to cover the set of policy selection requirements for the basic types of policies. It consists of the following elements:

- SPS-1. Scope of Security Policy Management
- SPS-2. Identification and Authentication Policy Selection
- SPS-3. Access Control Policy Selection
- SPS-4. TCB Interconnection Policy Selection
- SPS-5. Audit Policy Selection

Component SM_SPS.2. Inclusion of Availability Policies

This component extends SM_SPS.1 by adding the selection of specific availability policies for basic system services. It consists of the following elements:

- SPS-1. Scope of Security Policy Management
- SPS-2. Identification and Authentication Policy Selection
- SPS-3. Access Control Policy Selection
- SPS-4. TCB Interconnection Policy Selection
- SPS-5. Audit Policy Selection
- SPS-6. Availability Policy Selection

Component SM_SPS.3. Inclusion of Cryptographic Policies

This component extends SM_SPS.1 by adding the selection of specific cryptographic policies when required. This component consists of the following elements:

- SPS-1. Scope of Security Policy Management
- SPS-2. Identification and Authentication Policy Selection
- SPS-3. Access Control Policy Selection
- SPS-4. TCB Interconnection Policy Selection
- SPS-5. Audit Policy Selection
- SPS-7. Cryptographic Policy Selection

Component SM_SPS.4. Inclusion of Cryptographic Policies

This component extends both SM_SPS.2 and SM_SPS.3 by including both the selection of specific availability policies and cryptographic policies. This component consists of the following elements:

- SPS-1. Scope of Security Policy Management
- SPS-2. Identification and Authentication Policy Selection
- SPS-3. Access Control Policy Selection
- SPS-4. TCB Interconnection Policy Selection
- SPS-5. Audit Policy Selection
- SPS-6. Availability Policy Selection
- SPS-7. Cryptographic Policy Selection

It is envisioned that SM_SPS.1 will be used in the majority of profiles in which policies can be configured, whereas SM_SPS.2 and SM_SPS.3 will be used whenever either availability or cryptographic policies are needed and can be independently selected. Component SM_SPS.4 will be used whenever both availability and cryptographic policies are needed.

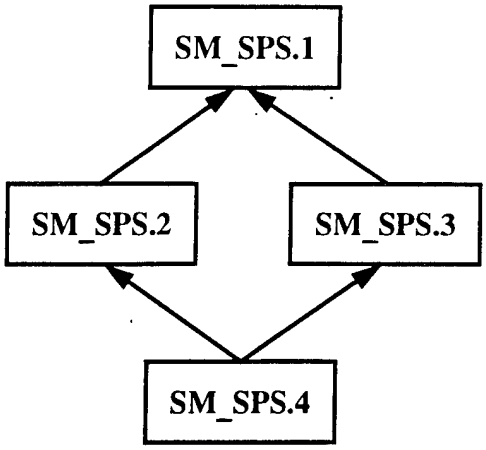


Figure 34. Component Relationships: Security Policy Selection

3. MANAGEMENT OF POLICY ATTRIBUTES

3.1 ELEMENTS

MPA-1. Policy Attributes Setting

(a) For each supported policy, administrative functions shall enable administrators to select, initialize, and modify the attributes of that policy that support the system security objectives.

(b) These attributes shall include subject registration, identification, authentication, system entry, and access control attributes for both the system and for individual users.

(c) These functions shall also enable system administrators to initialize and maintain the attributes of the audit policy.

Dependencies:

- Policy: SPS-1

MPA-2. Subject Registration Attributes

(a) Administrative functions within a system shall support subject registration (e.g., user, server, service machine registration) and establish accounts.

(b) These functions shall be able to perform the following:

1. **Define, display, and maintain subject registration and account attributes;**
2. **Import subject registration and account attributes from, and export to, local registries of autonomous host TCBs that are integrated within a distributed system; and**
3. **Define, display, and maintain security policy attributes of a subject and of the user's account (e.g., TCB privileges, groups, roles, system entry constraints such as time and location constraints).**

Dependencies:

- Uses: UA-1, DSE-3C
- Policy: MPA-1

MPA-3. Identification and Authentication Policy Attributes

(a) Each identification and authentication function shall allow the definition and maintenance of login policy attributes, including the session policy attributes

(e.g., if a password policy is selected, that policy shall allow the definition and maintenance of the password minimum and maximum lifetime, password length, and password complexity attributes).

(b) Policy attributes for a login session shall include the maximum period of interactive session inactivity, maximum login or session time, and threshold limit of successive unsuccessful attempts to login to the realm, with regard to each of the following selection criteria:

1. **Any port of entry (i.e., all ports),**
2. **A specific port of entry, and**
3. **A specific user identity or account.**

Dependencies:

- Policy: MPA-1, MPA-2, SPS-2

MPA-3A. Identification and Authentication Policy Attributes, Multiple Simultaneous Sessions

(a) Each identification and authentication function shall allow the definition and maintenance of login policy attributes, including the session policy attributes (e.g., if a password policy is selected, that policy shall allow the definition and maintenance of the password minimum and maximum lifetime, password length, and password complexity attributes).

(b) Policy attributes for a login session shall include the maximum period of interactive session inactivity, maximum login or session time, and threshold limit of successive unsuccessful attempts to login to the realm, with regard to each of the following selection criteria:

1. Any port of entry (i.e., all ports),
2. A specific port of entry, and
3. A specific user identity or account.

(c) The session policy attributes for a realm shall include those which limit the number of simultaneous login sessions per user identity on an authorization-attribute basis.

Dependencies:

- Policy: MPA-1, MPA-2, SPS-2

MPA-4. Access-Control Policy Attributes

(a) Administrative functions shall allow the definition and maintenance of access-control policy attributes, including those for determining the system security perimeter (e.g., for setting router-based access control within realm gateways).

(b) These functions shall be able to define and maintain attributes for user, group, role, and label-based access control (e.g., subjects' TCB privileges; system

entry constraints, such as time and location constraints; maximum number of groups; role hierarchy; label representation).

(c) These functions shall also be able to perform centralized distribution, review, and revocation of the security attributes of a subject.

Dependencies:

- Policy: MPA-1, SPS-3

MPA-5. Interconnection Policy Attributes

(a) Administrative functions shall allow the definition and maintenance of interconnection policy attributes for system security.

(b) These attributes shall include the following:

1. **Access control attributes for gateway routers,**
2. **Integrity and sensitivity ranges for each host TCB,**
3. **Inter-realm authentication path attributes (e.g., list of realms trusted by this realm, compromised realm list),**
4. **Limits placed on host TCB connectivity (e.g., types of hosts, numbers), and**
5. **Replication attributes for host TCBs supporting administrative functions.**

Dependencies:

- Policy: MPA-1, MPA-4, SPS-4

MPA-6. Audit Policy Attributes

(a) Administrative functions shall allow the definition and maintenance of audit policy attributes, including those for auditable events, event selection policy, conditional audit policy, and audit-trail management policies.

Dependencies:

- Policy: MPA-1, SPS-5

MPA-7. Availability Policy Attributes

(a) Administrative functions shall be provided to define and maintain resource control attributes, including resource quotas, quantity of resource limits, usage priorities, resource replication attributes, and timing thresholds.

(b) Administrative functions shall also be provided to define degraded-mode operations, re-configuration options, and contingency agreements for resource sharing in degraded-mode operations.

Dependencies:

- Policy: MPA-1, SPS-6

MPA-8. Cryptographic Policy Attributes

(a) Administrative functions shall allow the definition and maintenance of cryptographic policy attributes, including the following:

1. **Key management functions (e.g., key lifetime limits, key accountability attributes, and key-use restrictions), and**
2. **Cryptographic algorithms, and checksum and signature functions for data confidentiality and integrity.**

Dependencies:

- Policy: MPA-1, SPS-7

3.2 COMPONENTS

The components of this family refer to the need to instantiate each security policy selected for a system configuration by setting the policy attributes to specific values desired for that environment. The components parallel the policy selection components in the sense that they require the setting of the attributes for the policies included in the selection components. The second component has additional requirements for identification and authentication parameters. However, other components are possible for cases when only some policies offer a choice of attribute setting and others do not. The component of this family are rated based on the coverage of individual elements and the scope of policy attributes.

Component SM_MPA.1. Attribute Setting for Basic Policies

This component is intended to cover attribute setting requirements for the basic types of policies. It consists of the following elements:

- MPA-1. Policy Attributes Setting
- MPA-2. Subject Registration Attributes
- MPA-3. Identification and Authentication Policy Attributes
- MPA-4. Access-Control Policy Attributes
- MPA-5. Interconnection Policy Attributes
- MPA-6. Audit Policy Attributes

Component SM_MPA.1A. Extended Identification and Authentication Attributes

This component extends SM_MPA.1 by including additional requirements for setting identification and authentication attributes. It consists of the following elements:

- MPA-1. Policy Attributes Setting
- MPA-2. Subject Registration Attributes
- MPA-3A. Identification and Authentication Policy Attributes, Multiple Simultaneous Sessions
- MPA-4. Access-Control Policy Attributes
- MPA-5. Interconnection Policy Attributes
- MPA-6. Audit Policy Attributes

Component SM_MPA.2. Inclusion of Availability Attributes

This component extends SM_MPA.1 by adding the requirement for setting availability policy attributes. It consists of the following elements:

- MPA-1. Policy Attributes Setting
- MPA-2. Subject Registration Attributes
- MPA-3. Identification and Authentication Policy Attributes
- MPA-4. Access-Control Policy Attributes
- MPA-5. Interconnection Policy Attributes
- MPA-6. Audit Policy Attributes
- MPA-7. Availability Policy Attributes

Component SM_MPA.3. Inclusion of Cryptographic Attributes

This component extends SM_MPA.1 by adding the requirement for setting cryptographic policy attributes. It consists of the following elements:

- MPA-1. Policy Attributes Setting
- MPA-2. Subject Registration Attributes
- MPA-3. Identification and Authentication Policy Attributes
- MPA-4. Access-Control Policy Attributes
- MPA-5. Interconnection Policy Attributes
- MPA-6. Audit Policy Attributes
- MPA-8. Cryptographic Policy Attributes

Component SM_MPA.4. Inclusion of Availability and Cryptographic Attributes

This component extends both SM_MPA.2 and SM_MPA.2 by requiring the ability to set both availability and cryptographic policy attributes. This component consists of the following elements:

- MPA-1. Policy Attributes Setting
- MPA-2. Subject Registration Attributes
- MPA-3. Identification and Authentication Policy Attributes
- MPA-4. Access-Control Policy Attributes
- MPA-5. Interconnection Policy Attributes
- MPA-6. Audit Policy Attributes
- MPA-7. Availability Policy Attributes
- MPA-8. Cryptographic Policy Attributes

It is envisioned that SM_MPA.1 will be used in the majority of profiles in which policy attributes can be set, whereas SM_MPA.1A can be used for those profiles requiring

support for multiple, simultaneous user sessions. Components SM_MPA.2 and SM_MPA.3 will be used whenever availability or cryptographic policies are needed and their attributes can be independently set. Component SM_MPA.4 will be used whenever both availability and cryptographic policies are needed.

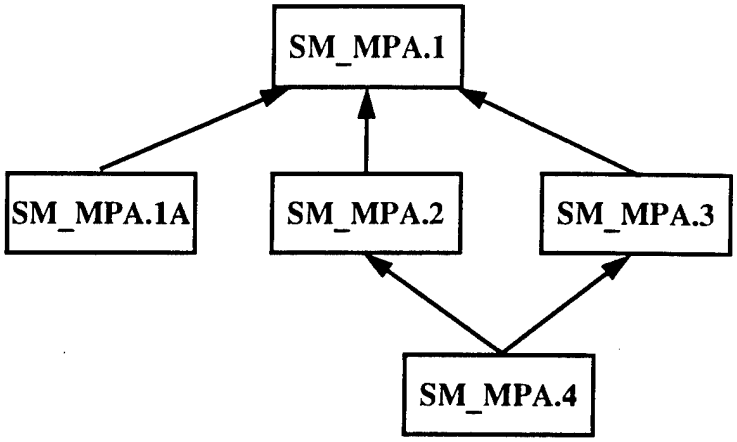


Figure 35. Component Relationships: Management of Policy Attributes

4. SEPARATION OF ADMINISTRATIVE ROLES

4.1 ELEMENTS

SAR-1. Basic Separation of Administrative Roles

(a) Administrative functions shall be able to perform the following:

1. **Define the administrative roles and their relationships (e.g., separation of duty, conflicts of interest, inclusion), and**
2. **Initialize system parameters to support administrative role separation (e.g., limit administrator login at designated hosts).**

Dependencies:

- Uses: PO-1A

SAR-2. Cryptographic-Domain Administrator Role

(a) The administrative roles shall include that of the cryptographic-domain administrator.

(b) This role shall be assumed by an authorized administrator performing cryptographic function initialization and management (e.g., cryptographic key and parameter entry, cryptographic key cataloging, audit functions, alarm resetting).

Dependencies:

- Uses: SAR-1, CDP-1

SAR-3. Cryptographic-Domain Maintenance Role

(a) The administrative roles shall include a maintenance role for the privileged maintenance interface.

(b) This role shall be assumed by an authorized administrator performing specific maintenance tests and obtaining interim results in order to service or repair functions of the cryptographic domain.

Dependencies:

- Uses: SAR-1, CDP-4

4.2 COMPONENTS

The components of this family are intended for use in profiles where strict separation of administrative roles is required. While the basic separation of administrative role component does not require any specific role, it does include a requirement for defining the relationships among the defined roles. In contrast, the second and third components require specific roles related to the management of the cryptographic domain to reflect the added concerns of current standards. The rating of the role separation components is based on the coverage of individual elements.

Component SM_SAR.1. Basic Separation of Administrative Role

This component includes the basic requirements for administrative role separation and consists of a single element:

- SAR-1. Basic Separation of Administrative Roles

Component SM_SAR.2. Separation of Cryptographic Administrator's Role

This component extends SM_SAR.1 by including the requirement for a separate cryptographic administrator role. It consists of the following elements:

- SAR-1. Basic Separation of Administrative Roles
- SAR-2. Cryptographic-Domain Administrator Role

Component SM_SAR.3. Separation of Cryptographic-Domain Maintenance Role

This component extends SM_SAR.2 by including the requirement for a separate cryptographic-domain maintenance role. It consists of the following elements:

- SAR-1. Basic Separation of Administrative Roles
- SAR-2. Cryptographic-Domain Administrator Role
- SAR-3. Cryptographic-Domain Maintenance Role

It is anticipated that component SM_SAR.1 will be included in all profiles where administrative role separation is desired. Component SM_SAR.2 is useful for environments where cryptographic functions are necessary for the protection of data communication, whereas component SM_SAR.3 is intended for environments where substantial control over the cryptographic operation is deemed necessary.

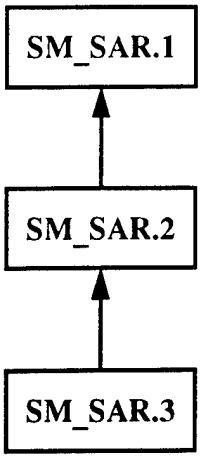


Figure 36. Component Relationships: Separation of Administrative Roles

5. SECURITY MANAGEMENT TOOLS

5.1 ELEMENTS

SMT-1. Administrative Tools

(a) The administrative functions shall include tools for distributed system initialization, recovery, maintenance, and audit, as well as tools to support individual host security management functions.

(b) Use of these tools shall require separate privileges available only to system administrators.

Dependencies:

- Uses: PO-1A, SI-3, EU-1, SC-2, SR-2, AM-4

SMT-1A. Administrative Tools for Distributed Systems

(a) The administrative functions shall include tools for distributed system initialization, recovery, maintenance, and audit, as well as tools to support individual host security management functions.

(b) Use of these tools shall require separate privileges available only to system administrators.

(c) Use of these tools shall not require a separate login to each host of the distributed system.

Dependencies:

- Uses: PO-1A, SI-3, EU-1, SC-2, SR-2, AM-4, AM-6

5.2 COMPONENTS

The components of this family include a basic element requiring that administrative tools be available to system administrators, and that access to these tools be restricted to administrative roles. In addition, a requirement for single login is added for use in distributed systems profiles.

Component SM_SMT.1. Basic Administrative Tools

This component consists of requirements for basic administrative tools and consists of a single element:

- SMT-1. Administrative Tools

Component SM_SMT.1A. Administrative Tools for Distributed Systems

This component consists of requirements for basic administrative tools for distributed systems and consists of a single element:

- SMT-1A. Administrative Tools for Distributed Systems

It is envisioned that component SM_SMT.1 will be used in most profiles that require separate tools for security management. Component SM_SMT.1A is intended for use in distributed systems supporting a single login.

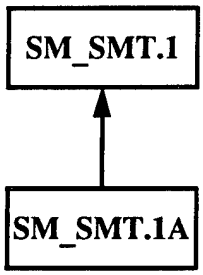


Figure 37. Component Relationships: Security Management Tools

BIBLIOGRAPHY

A. TRUSTED COMPUTING BASE

Gupta, S. and V. D. Gligor. 1992. Towards a Theory of Penetration-Resistant Systems. *Journal of Computer Security* Vol. 1/No. 2: pp. 133-158.

Gupta, S. and V. D. Gligor. 1992. Experience with a Penetration Analysis Method and Tool. In *Proceedings of the 15th National Computer Security Conference, Baltimore, MD, October 13-16, 1992*, pp. 165-183.

Hecht, M. S., M. E. Carson, C. S. Chandersekaran, R. S. Chapman, L. J. Dotterer, V. D. Gligor, W. D. Jiang, A. Johri, G. Luckenbaugh, and N. Vasudevan. 1987. Unix Without the Superuser. In *Proceedings of the USENIX Conference, Phoenix, AZ, June 8-12, 1987*, pp. 243-256. Berkeley, CA: USENIX Association.

International Telegraph and Telephone Consultative Committee (CCITT). 1988. *Message Handling Systems—Message Transfer System: Abstract Service Definition and Procedures*. Recommendation X.411. Geneva, CH: CCITT.

Purdue University. November 1988. *The Internet Work Program: An Analysis*. Purdue Technical Report CSD-TR-823. West Lafayette, IN: Purdue University.

U. S. Department of Defense, National Computer Security Center (NCSC). December 1991. *A Guide to Understanding Trusted Recovery in Trusted Systems*. NCSC-TG-022, Version 1. Fort George G. Meade, MD: NCSC. Also published in V. D. Gligor, *A Guide to Understanding Trusted Recovery*, Institute for Defense Analyses (IDA) Paper P-2307 (Alexandria, VA: IDA, September 1989).

B. IDENTIFICATION AND AUTHENTICATION

Abadi, M., M. Burrows, C. Kaufman, and B. Lampson, 1991. Authentication and Delegation with Smart-Cards. In *Proceedings of the International Conference on Theoretical Aspects of Computer Software (TACS), September 1991*, pp. 326-345. Berlin, DE: Springer-Verlag. Also published in Digital Equipment Corporation (DEC), Systems Research Center, Research Report 67 (Palo Alto, CA: DEC, October 1990).

Bellovin, S. M. and M. Merritt. 1990. Limitations of the Kerberos Authentication System. *Computer Communications Review* Vol. 20/No. 5 (October): 119-132.

Bellovin, S. M. and M. Merritt. 1992. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May 4-6, 1992*, pp. 72-84. Los Alamitos, CA: IEEE Computer Society Press.

Bellovin, S. M. and M. Meritt. 1993. Augmented Encrypted Key Exchange. In *Proceedings of the First ACM Conference on Computer and Communications Security, Fairfax, VA, November 3-5, 1993*, pp. 244-250. New York, NY: ACM Press.

Bird, R., I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. 1993. Systematic Design of a Family of Attack-Resistant Authentication Protocols. *IEEE Journal on Selected Areas in Communications* Vol. 11/No. 5 (June): pp. 679-693.

Bird, R., I. Gopal, A. Herzberg, P. Janson, and S. Kutten. 1995. The KryptoKnight Family of Light-Weight Protocols for Authentication and Key Distribution. *IEEE Transactions on Networking* Vol. 3/No. 1 (February): pp. 31-42.

Birrell, A. 1985. Secure Communication Using Remote Procedure Calls. *ACM Transactions on Computer Systems* Vol. 3/No. 1 (February): pp. 1-14.

Birrell, A., B. Lampson, R. Needham, and M. Schroeder. 1986. A Global Authentication Service without Global Trust. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, April 7-9, 1986*, pp. 223-230. Washington, DC: IEEE Computer Society Press.

Davis, D. and R. Swick. 1990. Network Security via Private-Key Certificates. *ACM Operating Systems Review* Vol. 24/No. 4 (October): pp. 64-67.

Denning, D. and G. Sacco. 1981. Timestamps in Key Distribution Protocols. *Communications of the ACM* Vol. 24/No. 8 (August): pp. 533-536.

Farrow, R. 1991. *UNIX System Security: How to Protect Your Data and Prevent Intruders*. Reading, MA. Addison-Wesley.

Feige, U., A. Fiat, and A. Shamir. 1987. Zero-Knowledge Proofs of Identity. In *Proceedings of the ACM Symposium on the Theory of Computing, New York, NY, May 25-27, 1987*, pp. 210-217. New York, NY: ACM Press.

Feldmeier, D. C. and P.R. Karn. 1989. UNIX Password Security—Ten Years Later. In *Advances in Cryptology—CRYPTO '89 Proceedings, Santa Barbara, CA, August 20-24, 1989*, pp. 44-63. Berlin, DE: Springer-Verlag.

Gasser, M. 1988. *Building a Secure Computer System.* New York, NY: Van Nostrand Reinhold.

Gasser, M. and E. McDermott. 1990. An Architecture for Practical Delegation in a Distributed System. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, May 7-9, 1990*, pp. 20-30. Los Alamitos, CA: IEEE Computer Society Press.

Gligor, V. D., S. W. Luan and J. N. Pato. 1993. Inter-Realm Authentication in Large Distributed Systems. *Journal of Computer Security* Vol. 2/Nos. 2-3: pp. 137-157.

I'Anson, C. and C. Mitchell. 1990. Security Defects in CCITT Recommendation X.509 The Directory Authentication Framework. *Computer Communications Review* Vol. 26/No. 2 (April): pp. 30-34.

International Organization for Standardization (ISO). 1988 (revised 1995). *Information Technology—Open Systems Interconnection—The Directory: Authentication Framework.* IS088 ISO/IEC 9594-8. Recommendation X.509. Geneva, CH: ISO. Also published as International Telecommunication Union, Telecommunication Standardization Bureau (ITU-T) Recommendation X.509 (Geneva, CH: ITU-T, [n.d.]).

International Telegraph and Telephone Consultative Committee (CCITT). 1988. *Information Technology—Open Systems Interconnection—The Directory: Overview of Concepts, Models, and Sevices.* Recommendation X.500. Geneva, CH: CCITT.

Internet Engineering Task Force (IETF). January 1993. *Request For Comment (RFC) 1411: Telnet Authentication: Kerberos Version 4.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1411.txt`. Four pages, ASCII format.

Internet Engineering Task Force (IETF). January 1993. *Request For Comment (RFC) 1412: Telnet Authentication : SPX.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1412.txt`. Four pages, ASCII format.

Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1416: Telnet Authentication Option.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1416.txt`. Seven pages, ASCII format.

Internet Engineering Task Force (IETF). September 1993. *Request For Comment (RFC) 1507: DASS - Distributed Authentication Security Service.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1507.txt`. One hundred nineteen pages, ASCII format.

Internet Engineering Task Force (IETF). September 1993. *Request For Comment (RFC) 1511: Common Authentication Technology Overview.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1511.txt`. Two pages, ASCII format.

Lamport. L. 1981. Password Authentication with Insecure Communication. *Communications of the ACM* Vol. 24/No. 11 (November): pp. 770-772.

Lampson, B., M. Abadi, M. Burrows, and E. Wobber. November 1992. Authentication in Distributed Systems: Theory and Practice. *ACM Transactions on Computing Systems* Vol. 10/No. 4: pp. 265-310.

Lampson, B. W. 1993. Authentication in Distributed Systems. In *Distributed Systems*, ed. S. Mullender. Reading, MA: Addison-Wesley.

Lomas, T. M. A., L. Gong, J. H. Saltzer, and R. M. Needham. 1989. Reducing Risks from Poorly Chosen Keys. In *Proceedings of the 12th ACM Symposium on Operating Systems Principles, Lichfield Park, AZ, December 3-6, 1989*, pp. 14-18. New York, NY: Association of Computing Machinery (ACM). Also published in *Operating Systems Review* Vol. 23/No. 5 (December 1989): pp. 14-18.

Massachusetts Institute of Technology (MIT). January 1992. *Kerberos Version 5, Revision 5*. Project Athena. Cambridge, MA: MIT.

Miller, S. P., B. C. Neuman, J. I. Schiller, and J. H. Saltzer. April 10, 1987. Kerberos Authentication and Authorization System, Section E.2.1 of the *Project Athena Technical Plan*. Cambridge, MA: Massachusetts Institute of Technology.

Molva, R., G. Tsudik, E. Van Herreweghen, and S. Zatti. 1992. KryptoKnight Authentication and Key Distribution System. In *European Symposium on Research in Computer Security. Toulouse, France, November 23-25, 1992*, pp. 155-174. Berlin, DE: Springer-Verlag.

Morris, R. and K. Thompson. 1979. Password Security: A Case History. *Communications of the ACM* Vol. 22/No. 11 (November): pp. 594-597.

Needham, R. and M. D. Schroeder. 1978. Using Encryption for Authentication in Large Network of Computers. *Communications of the ACM* Vol. 21/No. 12 (December): pp. 993-999.

Needham, R. and M.D. Schroeder. 1987. Authentication Revisited. *Operating Systems Review* Vol. 21/ No. 1 (January): p. 7.

Needham, R. M. 1993. Cryptography and Secure Channels. In *Distributed Systems*, ed. S. Mullender. Reading, MA: Addison-Wesley.

Neuman, B. C., J. I. Schiller, and J. G. Steiner. 1988. Kerberos: An Authentication Service for Open Network Systems. In *Proceedings of the USENIX Winter Conference, Dallas, TX, Februrary 9-12, 1988*, pp. 191-202. Berkeley, CA: USENIX Association.

Neuman, B. C. March 1991. *Proxy-Based Authorization and Accounting for Distributed Systems*. Technical Report 91-02-01. Seattle, WA: University of Washington.

Open Software Foundation (OSF). 1992. *Introduction to OSF DCE. Revision 1.0.* Cambridge, MA: OSF.

Open Software Foundation (OSF). June 1992. *OSF DCE SIG Request for Comment 3.0: Extending the OSF DCE Authorization System to Support Practical Delegation (Extended Summary)*. Available on the Internet, University of Massachusetts at Amherst, Project Pilgrim: URL `http://info.pilgrim.umass.edu/pub/osf_dce/RFC/rfc3.0.txt`. Fourteen pages, ASCII format.

Ottway, D. and O. Rees. 1987. Efficient and Timely Mutual Authentication. *Operating Systems Review* Vol. 21/ No. 1 (January): pp. 8-10.

Sollins, K. R. 1988. Cascaded Authentication. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, April 18-21, 1988*, pp. 156-165. Washington, DC: IEEE Computer Society Press.

Spafford, E. 1992. Observing Reusable Password Choices. In *Proceedings of the UNIX Security Symposium III, Baltimore, MD, September 14-16, 1992*, pp. 299-312. Berkeley, CA: USENIX Association.

Spafford, E. 1992. OPUS: Preventing Weak Password Choices. *Computers and Security* Vol. 11/No. 3: pp. 273-278.

Sun Microsystems, Inc. 1988. *Secure Networking*. Mountain View, CA: Sun Microsystems, Inc.

Sun Microsystems, Inc. 1988. *The Sun Yellow Pages Service*. Mountain View, CA: Sun Microsystems, Inc.

Sun Microsystems, Inc. [n.d.] *Secure Networking in the Sun Environment.* Mountain View, CA: Sun Microsystems, Inc.

Tardo, J. and K. Alagappan. 1991. SPX: Global Authentication Using Public Key Certificates. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, May 20-22, 1991*, pp. 232-244. Los Alamitos, CA: IEEE Computer Society Press.

U. S. Department of Commerce, National Bureau of Standards. September 1980. *Guideline of User Authentication Techniques for Computer Network Access Control.* FIPS PUB 83. Gaithersburg, MD: National Institute of Standards and Technology.

U. S. Department of Commerce, National Bureau of Standards. May 1985. *Password Usage.* FIPS PUB 112. Gaithersburg, MD: National Institute of Standards and Technology.

U. S. Department of Commerce, National Bureau of Standards. May 1985. *Computer Data Authentication.* FIPS PUB 113. Gaithersburg, MD: National Institute of Standards and Technology.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). October 1993. *Automated Password Generator.* FIPS PUB 181. Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). May 1994. *Digital Signature Standard.* FIPS PUB 186. Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). September 1994. *Guideline for the Use of Advanced Authentication Technology Alternatives.* FIPS PUB 190. Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). March 1995. *Standard for Public Key Cryptographic Entity Authentication Mechanisms.* FIPS Draft (March 13, 1995). Gaithersburg, MD: NIST.

C. SYSTEM ENTRY

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). March 1993. *Minimum Security Requirements for Multi-User Operating Systems.* Interagency Report 5153. Gaithersburg, MD: NIST.

D. TRUSTED PATH

Russell, D. and G. T. Gangemi Sr. 1991. *Computer Security Basics.* Sebastopol, CA: O'Reilly & Associates, Inc.

E. DATA CONFIDENTIALITY, F. DATA INTEGRITY

Blaze, M. 1994. Protocol Failure in the Escrowed Encryption Standard. In *Proceedings of the Second ACM Conference on Computer and Communications Security, Fairfax, VA, November 2-4, 1994*, pp. 59-67. New York, NY: ACM.

Internet Engineering Task Force (IETF). April 1992. *Request For Comment (RFC) 1321: The MD5 Message-Digest Algorithm.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1321.txt`. Twenty-one pages, ASCII format.

Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1421: Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1421.txt`. Forty-two pages, ASCII format.

Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1422.txt`. Thirty-two pages, ASCII format.

Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1423: Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1423.txt`. Fourteen pages, ASCII format.

Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1424.txt`. Nine pages, ASCII format.

Internet Engineering Task Force (IETF). September 1993. *Request For Comment (RFC) 1508: Generic Security Service Application Program Interface.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1508.txt`. Forty-nine pages, ASCII format.

Internet Engineering Task Force (IETF). September 1993. *Request For Comment (RFC) 1509: Generic Security Service API: C-Bindings.* Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1509.txt`. Forty-eight pages, ASCII format.

Jueneman, R. R., S. M. Matyas, and C. H. Meyer. 1984. Message Authentication with Manipulation Detection Codes. In *Proceedings of the 1983 Symposium on Security and Privacy, Oakland, CA, April 25-27, 1983*, pp. 33-54. Silver Spring, MD: IEEE Computer Society Press.

Jueneman, R. R., S. M. Matyas, and C. H. Meyer. 1985. Message Authentication. *IEEE Communications* Vol. 23/No. 9 (September): pp. 29-40.

Kent, S. 1993. Internet Privacy Enhanced Mail. *Communications of the ACM* Vol. 36/No. 8 (August): pp. 48-60.

Lambert, P. A. 1993. Layer Wars: Protect the Internet with Network Layer Security. In *Proceedings of the Privacy and Security Research Group Workshop on Network and Distributed System Security, San Diego, CA, February 11-12, 1993*, pp. 31-37.

Linn, J. and S. Kent. 1989. Privacy for DARPA Internet Mail. In *Proceedings of the 12th National Computer Security Conference, Baltimore, MD, October 10-13, 1989*, pp. 215-229. Fort George G. Meade, MD: National Computer Security Center.

Luan, S. W., and V. D. Gligor. 1990. On Replay Detection in Distributed Systems. In *Proceedings of the 10th International Conference on Distributed Computing Systems, Paris, France, May 28-June 1, 1990*, pp. 188-195. Los Alamitos, CA: IEEE Computer Society Press.

Merkle, R. 1990. One-Way Hash Functions and DES. In *Advances of Cryptology: Proceedings of Crypto '89, Santa Barbara, CA, August 20-24, 1989*, pp. 428-446. Berlin, DE: Springer-Verlag.

Merkle, R. 1990. A Fast Software One-Way Hash Function. *Journal of Cryptology* Vol. 3/ No. 1: pp. 43-58.

Meyer, C. H. and Matyas, S. M. 1982. *Cryptography: A New Dimension in Computer Data Security*. New York, NY: John Wiley and Sons.

Mitchell, C., M. Walker, and D. Rush. 1989. CCITT/ISO Standards for Secure Message Handling. *IEEE Journal on Selected Areas in Communications* Vol. 7/ No. 4 (May): pp. 517-524.

Miyaguchi, S., K. Ohta, and M. Iwata. 1991. Confirmation That Some Hash Functions Are Not Collision Free. In *Proceedings of Advances in Cryptology—EUROCRYPT '90: Workshop on the Theory and Application of Crytographic Techniques, Aarhus, Denmark, May 21-24, 1990*. Berlin, DE: Springer-Verlag.

Stubblebine, S. G. and V. D. Gligor. 1992. On Message Integrity in Cryptographic Protocols. In *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 4-6, 1992*, pp. 85-104. Los Alamitos, CA: IEEE Computer Society Press.

Stubblebine, S. G. and V. D. Gligor. 1993. Protecting the Integrity of Privacy-Enhanced Electronic Mail. In *Proceedings of the Privacy and Security Research Group Workshop on Network and Distributed System Security, San Diego, CA, February 11-12, 1993*, pp. 75-80. Washington, DC: Department of Energy.

U. S. Department of Commerce, National Bureau of Standards. January 1977. *Data Encryption Standard.* FIPS PUB 46. Gaithersburg, MD: National Institute of Science and Technology.

U. S. Department of Commerce, National Institute of Science and Technology (NIST). April 1993. *Secure Hash Standard.* FIPS PUB 180. Gaithersburg, MD: NIST.

Voydock, V. and S. Kent. 1983. Security Mechanisms in High Level Network Protocols. *ACM Computing Surveys* Vol. 15/No. 2: pp. 135-171.

X/Open. January 1994. *X/Open Preliminary Specification. Generic Security Service API (GSS-API) Base.* Reading, UK: X/Open Company Limited.

Ziv, J. and A. Lempel. 1977. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory* Vol. IT-23/No. 3 (May): pp. 337-343.

G. CRYPTOGRAPHIC SUPPORT

American National Standards Institute (ANSI). 1982. *Financial Institution Message Authentication.* American National Standard X9.9. New York, NY: ANSI.

American National Standards Institute (ANSI). 1985. *Financial Institution Key Management (Wholesale).* American National Standard X9.17. New York, NY: ANSI.

Balenson, D. 1985. Automated Distribution of Cryptographic Keys Using the Financial Institution Key Management Standard. *IEEE Communication* Vol. 23/No. 9 (September): pp. 41-46.

Beller, M. J., L-F. Chang, and Y. Yacobi. 1991. Privacy and Authentication on a Portable Communications System. In *Globecom '91, Phoenix, AZ, December 2-5, 1991*, pp. 1922-1927. New York, NY: IEEE.

Beller, M. J., Chang, L-F., Yacobi, Y. 1992. Security for Personal Communications Services: Public Key vs. Private Key Approaches. In *Proceedings of the 3rd IEEE International*

Symposium on Personal, Indoor, and Mobile Radio Communications Proceedings (PIMRC '92), Boston, MA, October 19-21, 1992, pp. 26-31. New York, NY: IEEE.

Beller, M. J., and Y. Yacobi. 1993. Fully-Fledged Two-Way Public Key Authentication and Key Agreement for Low-Cost Terminals. *Electronic Letters* Vol. 29/No. 11 (May): pp. 999-1001.

Biham, E. and A. Shamir. 1991. Differential Cryptanalysis of DES-like Cryptosystems. *Journal of Cryptology* Vol. 4/No.1: pp. 3-72.

Biham, E. and A. Shamir. 1992. Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOCI, and Lucifer. In *Advances in Cryptology—CRYPTO '91 Proceedings, Santa Barbara, CA, August 11-15, 1992*, pp. 156-171. Berlin, DE: Springer-Verlag.

Biham, E. and A. Shamir. 1993. Differential Cryptanalysis of the Full 16-Round DES. In *Advances in Cryptology—CRYPTO '92 Proceedings, Santa Barbara, CA, August 16-20, 1992*, pp. 487-496. Berlin, DE: Springer-Verlag.

Brickell, E., Denning, D., Kent, S., and Maher, D. July 28, 1993. *SKIPJACK Review Interim Report: The SKIPJACK Algorithm.* Washington, DC: Georgetown University, Office of Public Affairs.

Coppersmith, D. December 1992. *The Data Encryption Standard (DES) and Its Strength Against Attack.* IBM Research Report RC 18613 (81421). Yorktown Heights, NY: International Business Machines Corporation, T. J. Watson Research Center.

Davies, D. W. and Price, W. L. 1984. *Security for Computer Networks.* New York: John Wiley and Sons.

Davis, D., R. Ihaka, and P. Fenstermacher. 1994. Cryptographic Randomness from Air Turbulence in Disk Drives. In *Advances in Cryptology—Crypto '94, Santa Barbara, CA, August 21-15, 1994*, pp. 114-120. Lecture Notes in Computer Science, No. 839. Berlin, DE: Springer-Verlag.

Den Boer, B. 1988. Cryptanalysis of FEAL. In *Advances in Cryptology—EUROCRYPT 88, Davos, Switzerland, May 25-27, 1988*, pp. 293-299. Lecture Notes in Computer Science, Vol. 330. Berlin, DE: Springer-Verlag.

Den Boer, B. and A. Bosselaers. 1992. An Attack on the Last Two Rounds of MD4. In *Advances in Cryptology—Crypto '91 Proceedings, Santa Barbara, CA, August 11-15, 1991*, pp. 194-203. Berlin, DE: Springer-Verlag.

Denning, D. 1981. Timestamps in Key Distribution Protocols. *Communications of the ACM* Vol. 24/No. 8 (August): pp. 533-536.

Denning, D. 1982. *Cryptography and Data Security*. Reading, MA: Addison-Wesley.

Denning, D. 1983. Protecting Public Keys and Signature Keys. *IEEE Computer* Vol. 16/ No. 2 (February): pp. 27-35.

Desmedt, Y. and A. M. Odlyzko. 1986. A Chosen Text Attack on the RSA Cryptosystem and Some Discrete Logarithm Schemes. In *Advances in Cryptology—CRYPTO '85 Proceedings, Santa Barbara, CA, August 1985*, pp. 516-521. Lecture Notes in Computer Science, Vol. 218, ed. H. C. Williams. Berlin, DE: Springer-Verlag.

Diffie, W. and M. E. Hellman. 1976. A Critique of the Proposed Data Encryption Standard. *Communications of the ACM* Vol. 19/No. 3 (March): pp. 164-165.

Diffie, W. and M. E. Hellman. 1976. New Directions in Cryptography. *IEEE Transactions in Information Theory* Vol. IT-22/No. 6 (November): pp. 644-654.

Diffie, W. and M. E. Hellman. 1977. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *IEEE Computer* Vol. 10/No. 6 (June): pp. 74-84.

Diffie, W. and M. E. Hellman. 1979. Privacy and Authentication: An Introduction to Cryptography. In *Proceedings of the IEEE* Vol. 67/No. 3 (March): pp. 397-427.

Diffie, W. 1988. The First Ten Years of Public-Key Cryptography. In *Proceedings of the IEEE* Vol. 76/No. 5 (May): pp. 560-577.

El Gamal, T. 1985. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory* Vol. 31/No. 4 (July): pp. 469-472.

Fumy, W. and P. Landrock. 1993. Principles of Key Management. *IEEE Journal on Selected Areas in Communications* Vol. 11/No. 5 (June): pp. 785-793.

Hellman, M. E. 1978. An Overview of Public-Key Cryptography. *IEEE Transactions on Communications* Vol. 16/No. 6 (November): pp. 24-32.

Hellman, M. E. 1979. DES Will Be Totally Insecure Within Ten Years. *IEEE Spectrum* Vol. 16/No. 7 (July): pp. 32-39.

Hellman, M. E. and R. C. Merkle. 1981. On the Security of Multiple Encryption. *Communications of the ACM* Vol. 24/No. 7 (July): pp. 465-467.

IBM Corporation. 1991. *Common Cryptographic Architecture*. G321-#27. Manassas, VA: IBM Federal Systems.

International Organization for Standardization (ISO). October 1984. *Information Processing Systems—Data Communication—High-Level Data Link Control Procedure—Frame Structure*. ISO Standard 3309. 3d edition. Geneva, CH: ISO.

International Organization for Standardization (ISO). 1987. *Information Processing Systems: Modes of Operation for a 64-Bit Block Cipher Algorithm*. ISO Standard 8372. Geneva, CH: ISO.

Johnson, D. B., S. M. Matyas, A. V. Le, and J. D. Wilkins. 1994. The Commercial Data Masking Facility (CDMF) Data Privacy Algorithm. *IBM Journal of Research and Development* Vol. 38/No. 2 (March): pp. 217-226.

Jones, R. November 1982. Some Techniques for Handling Encypherment Keys. *ICL Technical Journal* Vol. 3/No. 2: pp. 175-188.

Kahn, D. 1967. *The Codebreakers: The Story of Secret Writing*. New York: Macmillan.

Konheim, A. 1981. *Cryptography: A Primer*. New York: John Wiley & Sons.

Le, A., S. Matyas, D. Johnson, and J. Wilkins. 1993. A Public Key Extension to the Common Cryptographic Architecture. *IBM Systems Journal* Vol. 32/No.3: pp. 461-485.

Matsui, M. 1994. Linear Cryptanalysis Method for DES Cipher. In *Proceedings of the EUROCRYPT '93, Lofthus, Norway, May 23-27, 1993*, pp. 386-397. Berlin, DE: Springer-Verlag.

Matyas, S. 1991. Key Handling with Control Vectors. *IBM Systems Journal* Vol. 30/No. 2: pp. 151-174.

Matyas, S., A. Le, and D. Abraham. 1991. A Key-Management Scheme Based on Control Vectors. *IBM Systems Journal* Vol. 30/No. 2: pp. 175-191.

Merkle, R. 1978. Secure Communication over Insecure Channels. *Communications of the ACM* Vol. 21/No. 4 (April): pp. 294-299.

Merkle, R. and M. Hellman. 1981. On the Security of Multiple Encryption. *Communications of the ACM* Vol. 24/No. 7 (July): pp. 465-467.

Miyaguchi, S., A. Shiraishi, and A. Shimizu. 1988. Fast Data Encryption Algorithm FEAL-8. *Review of Electrical Communications Laboratories* Vol. 36/No. 4: pp. 433-437.

Oorschot, P. and M. A. Wiener. 1991. Known-Plaintext Attack on Two-Key Triple Encryption. In *Proceedings of EUROCRYPT '90, Aarhus, Denmark, May 21-24, 1990*, pp. 318-325. Berlin, DE: Springer-Verlag.

Rivest, R. L., A. Shamir, and L. A. Adelman. 1978. Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM* Vol. 21/No. 2 (February): pp. 120-126.

Shamir, A. 1981. *On the Generation of Cryptographically Strong Pseudo-Random Sequences*. Rehovot, Israel: The Weizman Institute of Science, Department of Applied Mathematics.

Shimizu, A. and S. Miyaguchi. 1988. Fast Data Encipherment Algorithm FEAL. *Advances in Cryptography—EUROCRYPT 87, Amsterdam, Netherlands, April 13-15, 1987*, pp. 267-278. Lecture Notes in Computer Science, Vol. 304. Berlin, DE: Springer-Verlag.

U. S. Department of Commerce, National Bureau of Standards. April 1981. *Guidelines for Implementing and Using the NBS Data Encryption Standard*. FIPS PUB 74. Gaithersburg, MD: National Institute of Standards and Technology.

U. S. Department of Commerce, National Bureau of Standards. 1981. *DES Modes of Operation*. FIPS PUB 81. Washington, DC: National Institute of Standards and Technology.

U. S. Department of Commerce, National Bureau of Standards. August 1983. *Interoperability and Security Requirements for Use of the Data Encryption Standard in the Physical Layer of Data Communications*. FIPS PUB 139. Gaithersburg, MD: National Institute of Standards and Technology.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). April 1992. *Key Management Using ANSI X9.17*. FIPS PUB 171. Gaithersburg, MD: NIST.

U.S. Department of Commerce, National Institute of Standards and Technology (NIST). May 1993. *Security Requirements for Cryptographic Modules*. FIPS PUB 140-1 (Draft). Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). December 1993. *Data Encryption Standard*. FIPS PUB 46-2. Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). February 1994. *Escrowed Encryption Standard*. FIPS PUB 185. Gaithersburg, MD: NIST.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). April 1995. *Secure Hash Standard*. FIPS PUB 180-1. Gaithersburg, MD: NIST.

Wiener, M. 1990. Cryptanalysis of Short RSA Secret Exponents. *IEEE Transactions on Information Theory* Vol. IT-36/No. 3 (May): pp. 553-558.

Wiener, M. May 1993. Efficient DES Key Search. Technical Report TR-244. Ottawa, Canada: Carleton University, School of Computer Science. Also published in *Crypto '93, Santa Barbara, CA, August 22-26, 1993* (Berlin, DE: Springer-Verlag, 1994).

H. ACCESS CONTROLS

Bell, D. E. and LaPadula, L. J. October 1974. *Secure Computer Systems: Mathematical Foundations and Model.* M74-244. Bedford, MA: MITRE Corporation.

Biba, K. J. April 1977. *Integrity Considerations for Secure Computer Systems.* ESD-TR-76-372. Bedford, MA: USAF Electronic Systems Division.

Cheswick, W. and Bellovin, S. 1994. *Firewalls and Internet Security.* Reading, MA: Addison-Wesley.

Clark, D. D. and D. R. Wilson. 1987. A Comparison of Commercial and Military Computer Security Policies. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, April 27-29, 1987,* pp. 184-195. Washington, DC: IEEE Computer Society Press.

Clark, D. D. 1990. Policy Routing in Internetworks. *Internetworking: Research and Experience* Vol. 1/No.1 (September): pp. 35-52.

Curry, D. 1992. *UNIX System Security.* Reading, MA: Addison-Wesley.

Denning, D. May 1976. A Lattice Model of Secure Information Flow. *Communications of the ACM* Vol. 19/No. 5 (May): pp. 236-243.

Denning. P., ed. 1990. *Computers Under Attack: Intruders, Worms, and Viruses.* Reading, MA: Addison-Wesley.

Ferraiolo, D. F. and R. Kuhn. 1992. Role-Based Access Control. In *Proceedings of the 15th NIST-NSA National Computer Security Conference, Baltimore, MD, October 13-16, 1992,* pp. 554-563. Gaithersburg, MD: National Institute of Standards and Technology.

Garfinkel, S. and Spafford, F. 1991. *Practical Unix Security.* Sebastopol, CA: O'Reilly & Associates.

Gasser, M. 1988. *Building a Secure Computer System.* New York: Van Nostrand Reinhold.

Gligor, V. D., C. S. Chandersekaran, R. S. Chapman, L. J. Dotterer, M. S. Hecht, W. D. Jiang, A. Johri, G. L. Luckenbaugh, and N. Vasudevan. 1987. Design and Implementation of Secure Xenix. *IEEE Transactions on Software Engineering* Vol. SE13/No. 2 (February): pp. 208-221.

Internet Engineering Task Force (IETF). May 1989. *Request For Comment (RFC) 1102: Policy Routing in Internet Protocols*. Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1102.txt`. Twenty-two pages, ASCII format.

Internet Engineering Task Force (IETF). June 1993. *Request For Comment (RFC) 1478: An Architecture for Inter-Domain Policy Routing*. Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1478.txt`. Thirty-five pages, ASCII format.

Lampson, B. 1973. A Note on the Confinement Problem. *Communications of the ACM*, Vol. 16/No. 10 (October): pp. 613-615.

Lampson, B. 1974. Protection. *ACM Operating Systems Review* Vol. 8/No. 1 (January): pp. 18-24.

Mohammed, I. and D. M. Ditts. 1994. Design for Dynamic User Role-Based Security. *Computers and Security* Vol. 13/No. 8: pp. 661-671.

Nessett, D. M. 1987. Factors Affecting Distributed System Security. *IEEE Transactions on Software Engineering* Vol. SE-13/No. 2 (February): pp. 233-248.

Solms, S. H. von and I. VanderMerve. 1994. The Management of Computer Security Profiles Using a Role-Oriented Approach. *Computers and Security* Vol. 13/No. 8: pp. 673-680.

I. COVERT CHANNEL COUNTERMEASURES.

U. S. Department of Defense, National Computer Security Center (NCSC). November 1993. *A Guide to Understanding Covert Channel Analysis of Trusted Systems*. NCSC-TG-030, Version 1. Fort George G. Meade, MD: NCSC.

J. AUDIT

Denning, D. 1987. An Intrusion-Detection Mode. *IEEE Transactions on Software Engineering* Vol. SE-13/No. 2 (February): pp. 222-232.

Heberlein, L., B. Mukherjee, and K. Levitt. 1992. Internetwork Security Monitor: An Intrusion-Detection System for Large-Scale Networks. In *Proceedings of the 15th National Computer Security Conference, Baltimore, MD, October 13-16, 1992*, pp. 262-271.

Ilgun, K. 1993. USTAT: A Real-Time Intrusion Detection System for UNIX. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May 24-26, 1993*, pp. 16-28. Los Alamitos, CA: IEEE Computer Society Press.

Javitz, H. and A. Valdes. 1991. The SRI IDES Statistical Anomaly Detector. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May 20-22, 1991*, pp. 316-326. Los Alamitos, CA: IEEE Computer Society Press.

Tsai, C. R., V. D. Gligor, and M. S. Hecht. 1990. Potential Pitfalls of a Distributed Audit Mechanism. In *Proceedings of the European Unix Users Group (EUUG), Nice, France, October 22-26, 1990*, pp. 91-103. Buntingford, UK: EUUG.

K. AVAILABILITY

Gligor, V. D. 1984. A Note on the Denial-of-Service Problem. In *Proceedings of the Symposium on Computer Security and Privacy, Oakland, CA, April 25-27, 1983*, pp. 139-149. Silver Spring, MD: IEEE Computer Society Press.

Gligor, V. D. 1985. Denial-of-Service Implications for Computer Networks. In *Proceedings of the DoD Invitational Workshop on Computer Network Security, New Orleans, LA, March 19-22, 1985*, pp. 9-33 to 9-48. Fort George G. Meade, MD: Department of Defense Computer Security Center.

Gligor, V. D. 1986. On Denial of Service in Computer Networks. In *Proceedings of the International Conference on Data Engineering, Los Angeles, CA, February 5-7, 1986*, pp. 608-617. Washington, DC: IEEE Computer Society Press.

Millen, J. K. 1992. A Resource Allocation Model for Denial of Service Protection. In *Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 4-6, 1992*, pp. 137-147.

Millen, J. K. 1995. Denial of Service: A Perspective. *Dependible Computing for Critical Applications - 4*, pp. 91-108. Springer-Verlag Series on Dependable Computing and Fault-Tolerant Systems, Vol. 9, eds. F. Cristian, G. LeLann, and T. Lunt. Berlin, DE: Springer-Verlag.

Yu, C.-F. and V. D. Gligor. 1990. A Specification and Verification Method for Preventing Denial of Service. *IEEE Transactions on Software Engineering* Vol. SE-16/No. 6 (June): pp. 581-592.

L. SECURITY MANAGEMENT

Internet Engineering Task Force (IETF). April 1993. *Request For Comment (RFC) 1446: Security Protocols for Version 2 of the Simple Network Management Protocol (SNMPv2).*

Available on the Internet, InterNIC Directory and Database Services: URL `http://ds.internic.net/rfc/rfc1446.txt`. Fifty-one pages, ASCII format.

Levine, P. et al. 1990. *Moira, the Athena Service Management System*, draft. Cambridge, MA: Massachusettes Institute of Technology.

National Computer Security Center (NCSC). October 18, 1989. *Trusted Facility Management Guideline, Version 1*. NCSC-TG-015. Fort George G. Meade, MD: NCSC.

National Computer Security Center (NCSC). October 1992. *Trusted Facility Manual Guideline*. NSA/NCSC Guideline NCSC-TG-016. Fort George G. Meade, MD: NCSC.

Open Software Foundation (OSF), Inc. January 1992. *Distributed Computing Environment (DCE)—Design of the Security Services and Facilities, Version 1.0*. Cambridge, MA: OSF.

Stallings, W. 1993. *SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards*. Reading, MA: Addison-Wesley.

Tsai, C.-R., and V. D. Gligor. 1992. Distributed System and Security Management with Centralized Control. In *Proceedings of the 1992 EurOpen/Usenix Workshop, Jersey, UK, April 6-9, 1992*, pp. 137-146. Buntingford, UK: EurOpen.

OTHER SECURITY CRITERIA

Commission of the European Community. June 1991. *Information Technology Security Evaluation Criteria, Version 1.2*. Brussels, Belgium: Office for Official Publications of the European Community.

European Computer Manufacturers Association (ECMA). December 1989. *Security in Open Systems: Data Elements and Service Definitions*. Standard ECMA-138. Geneva, CH: ECMA.

Government of Canada, Communications Security Establishment. October 24, 1994. *Common Criteria for Information Technology Security Evaluations, Rationale, Parts 1, 2, and 3. Version 0.9*. CCEB-94/089 (Draft). Ottawa, Canada: Canadian System Security Centre.

Government of Canada, Communications Security Establishment. January 1993. *Canadian Trusted Computer Product Evaluation Criteria, Version 3.0e*. Ottawa, Canada: Canadian System Security Centre.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST), and the National Security Agency. December 1992. *Federal Criteria for Information Technology Security, Volumes I and II, Version 1* (Draft). Gaithersburg, MD: NIST.

U. S. Department of Defense (DoD). 1985. *Trusted Computer System Evaluation Criteria.* DoD 5200.28-STD. Washington, DC: U. S. Government Printing Office.

U. S. Department of Defense, National Computer Security Center (NCSC). July 1987. *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria.* NCSC-TG-005, Version 1. Fort George G. Meade, MD: NCSC.

U. S. Department of Defense, National Computer Security Center (NCSC). April 1991. *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria.* NCSC-TG-021, Version 1. Fort George G. Meade, MD: NCSC.

MISCELLANEOUS REFERENCES

Champine, G. A., D. E. Geer, and W. N. Ruh. 1990. Project Athena as a Distributed Computer System. *IEEE Computer* Vol. 23/No. 9 (September): p. 40.

Gasser, M., A. Goldstein, C. Kaufman, and B. Lampson. 1989. The Digital Distributed System Security Architecture. In *Proceedings of the 12th National Computer Security Conference, Baltimore, MD, October 10-13, 1989,* pp. 305-331.

National Research Council, System Security Study Committee. 1991. *Computers at Risk: Safe Computing in the Information Age.* Washington, DC: National Academy Press.

Saltzer J., D. Reed, and D. Clark. 1984. End-to-End Arguments in System Design. *ACM Transactions on Computing Systems* Vol. 2/No. 4 (November): pp. 277-288.

U. S. Department of Commerce, National Institute of Standards and Technology (NIST). November 1994. *Guideline for the Analysis of Local Area Network Security.* FIPS PUB 191. Gaithersburg, MD: NIST.

U. S. Department of Defense, National Computer Security Center (NCSC). October 21, 1987. *Glossary of Computer Security Terms.* NCSC-TG-004, Version 1. Fort George G. Meade, MD: NCSC.

GLOSSARY[1]

AND-chained identities. A conjunction of subject identities.

asymmetric cryptographic algorithm. Mathematical formulation which generates a public- and private-key pair used to encrypt and decrypt data. The public key is used to encrypt the message and the private key is used to decrypt it, ensuring that only the intended receiver can read the message.

asymmetric keys. A pair of keys, one public and one private, that is generated through the execution of an asymmetric cryptographic algorithm used in the encryption and/or decryption process.

authentication authority. An agent acting on the behalf of a given subject that is a source for certified identities.

authentication channel. A communication link that is established for the purpose of providing a secure means of establishing a subject's identification.

authentication data. Security-relevant data associated with authentication such as passwords, secret or private keys.

authentication paths. Indicates which of the TCBs and trusted authentication authorities of a distributed system are used to perform authentication.

bypass rate. The rate of the diversion of bypass data around the cryptographic function.

bypass data. Part of a message that is diverted around the cryptographic function.

channel. An information transfer path that can be defined as a series of encrypted messages.

channel data. Information that is included in a message packet which is sent across a channel.

channel revocation. The release of a communications link.

---

[1] The definitions provided in this Glossary were obtained from the references listed in the Bibliography.

ciphertext. Enciphered information.

communication channel. The physical media and devices which provide the means for transmitting information from one part of a distributed system to another.

communication processors and media. The physical hardware that is used in a distributed system.

communication protocol. The rules or conventions that are used to transfer data across a communication line which provides a communication service (e.g., TCP/UDP/IP, SNA).

compound subjects. A subject that is the concatenation of one or more subjects that is represented as either a conjunction of subjects or as a delegation chain.

conjunctions of subjects. A subject consisting of several subject identities. A type of a compound subject.

confidentiality. The property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

cryptographic checksum. A check value that is derived by performing a cryptographic transformation on a data unit. The derivation of the checkvalue may be performed in one or more steps and is a result of a mathematical function of the key and data unit. It is usually used to check the integrity of a data unit.

cryptography. The discipline that embodies the principles, means, and the methods for the transformation of data in order to hide its information content, prevent its undetected modification, and/or prevent its unauthorized use.

cryptographic algorithm. A mathematical formulation which either produces keys or performs encryption and/or decryption.

cryptographic domain. The totality of protection mechanisms consisting of hardware, software, and firmware that securely provides a trusted realm for cryptography that is isolated and non-circumventable.

cryptographic functions. Functions that ensure channel separation and data protection.

cryptographic token. Used to pass securely the identity of the client for whom the token was issued to a server. Also called a ticket.

crypto-ignition key (CIK). A device or electronic key that is used to unlock the secure mode of crypto-equipment.

data confidentiality functions. Functions that ensure that sensitive data are not disclosed in an unauthorized manner while being transmitted between trusted hosts via communication channels.

data integrity. The property that data have not been altered or destroyed in an unauthorized manner.

data integrity functions. Functions that ensure that message data are not modified in an undetectable manner while being transmitted between the trusted hosts of a distributed system via communication channels.

decipherment. The reversal of a corresponding reversible encipherment.

decryption. Generic term encompassing decoding and deciphering.

delegation. When one subject hands off some or all of its authority to another subject, i.e., A speaks for B if the fact that subject A says something means that subject B says the same thing.

delegation chains. A list of all the subjects that are acting on behalf of a particular subject. Each subject in the chain enables and/or disables the delegation of its identity. Delegation chains must be able to preserve the distinction between the identity of the original subject and that of the delegates. A type of a compound subject.

digital signature. Process which operates on a message to assure message source authenticity and integrity, and source non-repudiation.

distributed system. A collection of nodes connected by communication links to one or more networks, which participates in the routing of messages within these networks. It differs from a network in that the existence of autonomous computers is handled transparently by the distributed system.

domain. A basic unit of operation and administration—a group of users, systems, and resources that typically have a common purpose and share common services. Also known as a cell or a realm.

encipherment. The cryptographic transformation of data to produce ciphertext. *See also:* **cryptography.**

encrypted key. A private key that has been enciphered so that it can be sent over a communications link securely.

encryption. Generic term encompassing enciphering and encoding.

escrow authorities. An organization which has been invested with the right to store archived keys.

expired key. A key whose lifetime is no longer valid.

gateway. A device connecting two or more networking systems, which mediates the transfer of information from one system to another.

integrity. *See:* **data integrity**.

interconnection policy. Consists of a set of rules that define whether secure channels may be established between trusted hosts of a security perimeter and among different security perimeters. It also defines the type of trusted channels (e.g., for confidentiality only, integrity and availability, or authentication only) that can be established subject to these interconnection policies.

key. A sequence of symbols that controls the operations of encryption and decryption.

key attributes. Control fields consisting of security information which is associated with the key.

key component. One of at least two parameters having the format of a cryptographic key that are input to the Boolean exclusive-OR function to form a cryptographic key.

key distribution. A manual or automated process of securely assigning keys to be used between a user and a server.

key escrow. The process of storing keys to be archived with an escrow authority.

key generation. The propagation of a key that results from the execution of a cryptographic algorithm.

key installation. The inserting of a cryptographic algorithm into a cryptographic domain.

key lifetime. Specifies the times that a key is valid and its expiration date. All keys should have a finite lifetime.

key management. The generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy.

key space. Range of possible values of a key.

maintenance key. A key intended only for off-the-air in shop use.

message authentication. A procedure that lets communicating parties verify that the messages they receive are authentic.

message authentication code (MAC). A data element associated with an authenticated message which allows a receiver to verify the integrity of the message.

message-origin authentication. A procedure that, when established between two or more communicants, allows each party to verify that the received messages are genuine.

message stream. The transferring of data across an end-to-end communication link.

mutual authentication. Provides mutual assurance regarding the identity of subjects and objects. For example, a system needs to authenticate a user, and a user needs to authenticate that the system is genuine.

network link. A transmission line that is used to transfer data.

non-repudiated authentication. A process, which cannot be subsequently refuted, to identify each entity trying to gain access via communication links and assert that each entity is genuine.

packet. A message data unit.

plaintext. Unencrypted information.

port. A logical or physical identifier that a computer uses to distinguish different terminal input and output data streams.

private key. A secret key generated through the use of a symmetric cryptographic algorithm that is shared and known only to the communicating parties.

protocol control information. Reserved fields in a message packet that contains sequence numbers, acknowledgments, checksums, timestamps, or other information that is necessary for the correct execution of the protocol.

public key. A key generated through the use of an asymmetric algorithm that is publicly available. It makes up one half of a public- and private-key pair that is used for encryption and decryption.

realm. *See:* **domain.**

receiver. Subject reading from a communication channel.

repudiation. Denial by one of the entities involved in a communication of having participated in all or part of the communication.

router. A device used to connect two networks.

routing. The process of transferring packets between a source and a destination that may involve packets making multiple hops before reaching the destination.

secret key. A conventional key that is generated through the use of a symmetric cryptographic algorithm which is shared and known only to the communicating parties.

secure channel. *See:* **trusted channel.**

secure distributed system. A distributed system consisting of a set of TCBs and/or realms that are connected through secure channels, contained within one or more security perimeters, and subject to interconnection policies and constraints placed on one or several of the security perimeters.

security perimeter. The interface between the network, the hosts, and the gateways of a distributed system. It represents a partition of a distributed system product that delimits the scope of administrative control over the product and application resources (e.g., hosts, communication gateways), and the scope of security policies being enforced by a single, centralized administrative organization.

security policy. The set of criteria for the provision of security services.

seed key. The initial key that is used to start an updating or key generating process.

sender. A subject writing to a channel.

session key. A temporary key that is generated when a connection is established and is used for the duration of a single communication session.

signature generation. The execution of an asymmetric key algorithm that is used to sign messages.

simple subject. *See:* **subject.**

sliding time window. A process for transmitting data in two directions over one communication channel where, at any given instant of time, the sender maintains a list of consecutive sequence numbers corresponding to frames it is permitted to send, and the receiver maintains a list corresponding to frames it is permitted to accept.

smart card. A small computer in the shape of a credit card. Typically used to identify and authenticate its bearer.

split knowledge. The separation of data or information into two or more parts, with each part constantly kept under the control of separate authorized individuals or teams, so that no one individual or team has knowledge of the total data.

subject. Active entity in an information technology product, generally in the form of a process or device, that causes information to flow among objects or changes the system state.

symmetric cryptographic algorithm. A mathematical formulation that generates a single private key used to encrypt and decrypt data. This key is shared and known only among the communicating parties.

symmetric keys. A single private key that is generated through the execution of a symmetric cryptographic algorithm and is used for encryption and decryption. Also referred to as a conventional key.

threshold. A level below which activities can proceed in a correct or secure manner.

token-based card. A pocket-sized computer that can participate in a challenge-response authentication scheme.

trusted channel. An information transfer path that contains the channel authentication, availability, confidentiality, and integrity properties, in which the set of all possible senders can be known to the receivers, or the set of all possible receivers can be known to the senders, or both.

LIST OF ACRONYMS

AT&T	Atlantic Telephone and Telegraph, Inc.
CA	Certificate Authority
CRL	Certificate Revocation List
DES	Data Encryption Standard
DoD	Department of Defense
EDI	Electronic Data Interchange
GPS	Global Positioning System
I&A	Identification and Authentication
IDA	Institute for Defense Analyses
ID	Identifier
IT	Information Technology
MTA	Message Transfer Agent
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTP	Network Time Protocol
OID	Object Identifier
PIN	Personal Identification Number
RFC	Request For Comment
RPC	Remote Procedure Call
RSA	Rivest, Shamir, and Adleman
TBD	To Be Determined
TCB	Trusted Computing Base
TCSEC	Trusted Computer Systems Evaluation Criteria

APPENDIX A. THE REFERENCE MONITOR CONCEPT

The concept of the reference monitor, "which enforces the authorized access relationships between subjects and objects of a system," was introduced by the Computer Security Technology Planning Study, conducted by James P. Anderson & Co., in October of 1972. The reference monitor concept was found to be an essential element of any product that must demonstrably implement an access control policy. The Anderson report listed three design requirements of the reference validation mechanism which is "an implementation of the reference monitor concept that validates each reference to data or programs by any user (program) against a list of authorized types of reference for that user." These requirements are as follows:

a. The reference validation mechanism must be tamperproof.

b. The reference validation mechanism must always be invoked.

c. The reference validation mechanism must be small enough to be subject to analysis and tests, the completeness of which can be assured.

Early examples of the reference validation mechanism were known as *security kernels*. Security kernels typically support the three reference monitor requirements listed above. However, most commercially available systems do not implement reference validation mechanisms (e.g., security kernels) largely because their design and implementation do not fully satisfy requirement (*c*). General-purpose systems do not support security kernels, and their TCB generally includes key elements of the operating system and may include all of the operating system. In embedded systems, the security policy may deal with objects in a way that is meaningful at the application level rather than at the operating system level. Thus, the protection policy may be enforced in the application software rather than in the underlying operating system. The TCB will necessarily include all those portions of the operating system and application software essential to the support of the policy.

Note that, as the amount of code in the TCB increases, it becomes harder to be confident that the TCB enforces the reference monitor requirements under all circumstances. This suggests that, to demonstrably satisfy requirement (*c*) of the reference validation

mechanism, the selection of functions to be designed within the product must be governed by the ability to completely analyze and test the reference validation mechanism. If state-of-the-art formal methods are required for complete analysis and test of a product, the product functions that become part of the reference validation mechanism will, by necessity, be limited in scope. For example, functions that support a wide selection of devices and access methods may not be supported. Also, access control functions whose design and/or implementation by the reference validation mechanism are not, or cannot be, completely analyzed may limit the degree of assurance that can be obtained. Thus, requirement (*c*) establishes a dependency of the access control functions on the design, specification, and verification disciplines used in analysis and testing.

The concept of the reference monitor, and its implementation via the reference validation mechanism, plays the key role in supporting a wide variety of access control policies. However, the role of the reference monitor concept in other security policy areas is, by definition, limited. For example, the reference validation mechanism is not intended to implement identification and authentication policies (e.g., policies governing the choice of password complexity, strength of the encryption functions). Nor is the reference validation mechanism intended to implement availability policy (i.e., resource allocation and fault-tolerance). Furthermore, the reference validation mechanism plays an important but incomplete role in establishing the penetration resistance of a TCB. Although the reference validation mechanism itself must be penetration resistant by virtue of requirements (*a*) and (*b*), penetrations caused by weak authentication or availability functions, and penetrations of privileged processes of the TCB that are not part of the reference validation mechanism, cannot be prevented by a (penetration-resistant) reference validation mechanism.

APPENDIX B. DEFINING ACCESS CONTROL POLICIES

There are two aspects to consider when defining access control policies for trusted computing systems. The first is the definition of a basic access control policy, and the second is the composition of two or more basic policies. The issues surrounding these two policy definition considerations are discussed in individual sections of this appendix.

Defining a Product Policy

Access control policies can be characterized in terms of three functional families of requirements, namely (1) definition of subject and object policy attributes, (2) administration of the policy attributes, and (3) authorization of subject access to objects (which also includes authorization for subject and object creation and destruction, and object encapsulation access). These three families, defined in the following paragraphs, can be used to characterize a wide variety of security policies, including traditional discretionary and non-discretionary policies. The intent of characterizing all security policies in terms of these three families is to provide a general set of requirements applicable to all policies, regardless of the aim of those policies and regardless of the kinds of objects controlled by those policies. These requirements provide the developers of protection profiles with a template for access control policies to be used in the definition of individual policies, without imposing any specific constraints on policy or on the kinds of objects involved.

Since individual policies will follow this template, combinations of policies will also be defined in terms of the three families. Whenever multiple policies are supported, these families define the composition of policies and how the policies are enforced (e.g., subject and object type coverage, precedence of enforcement).

Within a policy specification, requirements can be stated as different sets of rules. These rules define the properties of each policy. Access control policy families may include requirements that may not be applicable to some policies. In such cases, the individual requirement shall be designated as non-applicable in the definition of the policy. For example, the transitive distribution of permissions applies primarily to discretionary policies. Consequently, attribute administration rules of non-discretionary controls may not include

conditions for transitive distribution and revocation, and these conditions will be designated as non-applicable to a specific non-discretionary policy. Similarly, discretionary policies may not necessarily control access to object status variables (e.g., existence, size, creation, access and modification time, locking state). Hence, the rules or conditions specifying such controls may be designated as non-applicable in specific discretionary policies.

Some families may also include requirements that may not be applicable to some types of objects. In such cases, the individual requirement that is applicable to that type of object will be specified separately. The intent of providing per-type access policy specifications is to capture the access control needs of a particular type of object without imposing impractical or meaningless policy constraints. For example, user-oriented rules for access-right administration need not be imposed on objects that cannot, and are not intended to, store user data. Requiring distribution and revocation conditions (e.g., transitive, temporal, time dependent, and/or location dependent) for a discretionary policy on interprocess communication objects such as semaphores and sockets or on publicly accessible objects such as bulletin boards would be both impractical and unnecessary. However, when per-type specifications are used, the totality of the per-type rules and conditions must be shown to support the policy properties.

Definition of Policy Attributes. A policy specification must define the subject and object attributes required by that policy, and must identify the context-dependent policy attributes. Subject attributes may include user-dependent credentials (e.g., user identifier, group or role identifier(s), confidentiality or integrity levels, access time intervals, access location identifier, realm identifier), as well as user-independent credentials (e.g., system privileges allowing the invocation of TCB functions unavailable to unprivileged subjects). Object attributes may include user-dependent, policy attributes (e.g., distinct object permissions for different users), as well as user-independent attributes (e.g., secrecy or integrity privileges accessible only to privileged processes). Finally, context-dependent policy attributes may include the current time, group definitions, and/or a level indicating whether an emergency is in progress.

If multiple policies are supported, the rules for defining subject and object attributes must partition these attributes on a policy basis.

Administration of Policy Attributes. A policy specification must include rules for maintaining the subject and object attributes. The attribute maintenance rules determine the conditions under which a subject can change its own attributes as well as those of other subjects and objects. These conditions define whether a subject is authorized to modify a policy

attribute and may not rely on those used in the authorization of subject access to objects (discussed on page B-4). Otherwise, a cyclic dependency may arise between the requirements of policy attribute administration and those of authorization of subject access to objects. The attribute maintenance rules also define the attributes for subject or object import or export operations.[1]

As an example of attribute maintenance rules, consider those rules that determine what subjects have the authority to distribute, revoke, and review policy attributes for specific subjects and objects, and the conditions under which these actions can be performed. The distribution and revocation rules determine which of the following conditions are enforced.

a. Selectivity: distribution and revocation can be performed at the individual attribute level, such as user, group, role, permission, privilege, security or integrity level.

b. Transitivity: a recipient of a permission from an original distributor can further distribute that permission to another subject, but when the original distributor revokes that permission from the original recipient, then the subject which received that permission from the original recipient will also have it revoked.

c. Immediacy: the effect of the distribution and revocation of policy attributes should take place within a specified period of time.

d. Independence: two or more subjects can distribute or revoke policy attributes to the same subject independent of each other.

e. Time dependency: the effect of the distribution and revocation of policy attributes must take place at a certain time and must last for a specified period of time.

f. Location dependency: the distribution and revocation of policy attributes must take place at a certain location.

The review rules determine which of the following two kinds of review are supported and also impose conditions constraining the review of attributes.

---

[1] U. S. Department of Commerce, National Institute of Standards and Technology (NIST), and the National Security Agency. December 1992. *Federal Criteria for Information Technology Security, Volumes I and II, Version 1*, Chapter 7–Construction of Protection Profiles (Draft). Gaithersburg, MD: NIST.

a. Per-object review: for an object, list all (or a specified class of) attributes that govern the relationship between that object and a specified set of subjects that may directly or indirectly access that object.

b. Per-subject review: for a subject, list all (or a specified class of) policy attributes which govern the relationship between that subject and a specified set of objects that the subject may directly or indirectly access.

The imposed conditions for allowing the review of attributes determine, in particular, which users of an object may discover which users have access to that object, as well as what subjects may be used to access that object.

The coverage of attribute-review rules is specified in terms of the kinds of objects and subjects to which they apply. If different rules and conditions apply to different subjects and objects, the totality of these rules must be shown to support the defined control objectives. If a composition of several policies is to be supported, attribute administration must be composed.

Authorization of Subject Access to Objects. A subject's access to an object consists of invoking an action on a set of objects. The subject's access to the object can be thought of as a request to access that object. Examples of actions include invocations of TCB commands, function calls, processor instructions, protected subsystems, and transactions. An action may have separate policy attributes from those of the issuer of the reference. For example, invocations of transactions and protected subsystems (which encapsulate objects) will generally include policy attributes that differ from those of their invokers. In contrast, other actions such as invocations of individual processor instructions, TCB function calls, some TCB commands, and applications programs are prohibited from using policy attributes, such as identity, group, role, or secrecy and integrity levels, that differ from those of their invoker. Policy attributes involved in rules for deciding access authorization are referred to as "access control" attributes.

The rules for authorizing subject access to objects are defined in terms of (1) the subject's authorization to an action, (2) the action authorization to one or more objects, and (3) the subject's authorization to one or more objects. These rules are based on the policy attributes defined for subjects and objects. The rules are defined either on *{subject, action}* and *{action, object(s)}* tuples or on *{subject, action, object(s)}* triples, depending upon the specified policy. The authorization rules specify the authorization scope and granularity in terms of (1) resources containing one or more objects, (2) individual subjects and objects,

(3) the subject and object policy attributes, and (3) the subject and object status attributes (e.g., existence, size, creation, access and modification time, lock status). The authorization rules also specify whether delegated authorization is allowed (i.e., authorization of a subject access performed on behalf of other subjects, using combined-subject attributes).

The coverage of the authorization rules is specified in terms of the types of objects and subjects to which they apply. If different rules apply to different subjects and objects, the totality of these rules is shown to support the defined policy properties. If multiple policies are supported, these rules define the composition of policies and how the authorization conditions are enforced (e.g., subject and object type coverage, order of enforcement)

The rules for authorizing the creation and destruction of subjects and objects must be defined. These rules impose the following conditions under which subjects and objects can be created and destroyed.

a. Creation and destruction authorization: the authorization of specific subjects to create and destroy a subject or an object and with what attributes.

b. Object reuse: the revocation of all authorizations to the information contained within a storage object prior to initial assignment, allocation, or reallocation of that storage object to a subject from the TCB's pool of unused storage objects; no information, including encrypted representations of information, produced by a prior subject's actions should be available to any unauthorized subject.

c. Space availability: the capacity and presence of storage space shall be available for the creation of a subject or object.

d. Definition of default attributes: subject attributes and the default values and rules for inheriting object attributes, if any, shall be defined.

The authorization of access to encapsulated objects specifies that a subject's access to objects be constrained in such a way that (1) all accesses to these objects occur via access to a logically and/or physically isolated set of subjects that protect these objects from more general forms of access, with each subject having a unique protected entry point; and (2) confinement of this set of protecting subjects is such that these subjects cannot access any other objects and cannot give away access to the objects they protect.

Discretionary encapsulation allows individual (privileged and unprivileged) users to create protected subsystems and to set access to them at their own discretion (e.g., using well-known discretionary access control mechanisms). Non-discretionary encapsulation

uses logical and/or physical domains (and possibly security levels) to enforce encapsulation at the product level (i.e., by system administrators as opposed to at the discretion of the creator of the protected subsystem). The traditional Department of Defense (DoD) mandatory policies may be useful for encapsulation in some environments. For example, one could use DoD mandatory policies to encapsulate a protected subsystem by reserving a sublattice of compartments for the programs and data objects of that subsystem. (Some trusted database management systems use this approach for the support of per-client database management system servers. The server(s) and database objects are encapsulated in a reserved sublattice of the TCB). Note that both discretionary and non-discretionary encapsulation can involve the use of surrogate subjects to protect the entry points to protected subsystems.

The rules for object encapsulation must be defined whenever object encapsulation is supported. The rules for object encapsulation constrain (1) access authorization to encapsulated objects (i.e., a subject access to an object can take place only if the subject invokes another subject that performs the requested action on the object using additional authorizations associated with the encapsulation); (2) application-level encapsulation (i.e., they define conditions for the creation of encapsulated subsystems); and (3) invocation of encapsulated subsystems.

Composition of Access Control Policies within a Product

Many of the access control policies supported by a product represent a composition of two or more basic access control policies. The need to compose basic policies arises for at least two reasons. First, to extend the range of an information technology (IT) product's protection applicability, new applications subsystems or individual functions may be added to a TCB. These subsystems and functions may support different basic access control policies from those supported by the original TCB. These different policies must be composed with policies of the original TCB. Second, to support new system or organizational policies, functions implementing new basic access policies are required to be added to a product's TCB. These new access control policies must also be composed with the existing policies to enable the implementation of the protection objectives of an organization.

The composition of access control policies within a product adds new requirements to the definition of product access control policies. For example, whenever trusted subsystems or functions that extend the TCB are added to support new policies, it must be ensured that existing TCB functions cannot be used to access the new subjects and objects in an unauthorized way, and that the new subsystems and functions cannot be used to access

the currently existing subjects and objects in an unauthorized way. Also, whenever multiple policies are composed within the same TCB and refer to the same set of subjects and objects, it must be determined that the composition of access control policies is consistent with the overall TCB protection policy and does not introduce new vulnerabilities.

The composition of access control policies within an access control family also requires that both the individual access control policies and their rules for composition be completely defined (i.e., for each element of the defined policy, a corresponding set of rules must establish the completeness of the composition).

Composition of Discretionary and Non-Discretionary Policies. A typical example of access control policy composition within the same IT product TCB is provided by the addition of a non-discretionary access control policy (e.g., the DoD mandatory policy) to a TCB that originally supports only a discretionary policy. The composition rules for the resulting TCB access control policy require that (1) both the mandatory and discretionary authorization rules be enforced on every subject and object protected by discretionary controls, and (2) the access issued by the enforcement modules of the discretionary policy be subject to the mediation specified by the mandatory rules. This precedence of enforcement is important whenever the exceptions returned by the enforcement of the two sets of rules are different. The reason is that if non-identical exceptions are returned by the two sets of rules, new covert channels may appear that would not appear had only the mandatory rules be enforced. These covert channels would violate the intent of the mandatory secrecy policy.

Other examples of policy composition within the same TCB include those in which the DoD mandatory secrecy policy and a mandatory integrity policy are supported. This composition might imply (1) that both the mandatory authorization rules be enforced on every subject and object access, and (2) that the controlled sharing rules of the two mandatory policies must be compatible with each other. Compatibility of these rules would imply, for example, that the secrecy and integrity upgrade conditions must not introduce covert channels that otherwise would not exist when the individual policies were used separately.

Composition by Policy Partitioning. A typical example of policy partitioning appears when a subsystem implementing its own access control policy is integrated within an operating system TCB. (An alternate way of integrating such a subsystem in a trusted operating system is illustrated in the following discussion of TCB policy subsetting). Such subsystem integration is fairly common in database management systems and products. Since these subsystems implement their own policies, which generally differ from those of

the operating system, the composition must ensure that neither the operating system nor the database subsystem interfaces of the same TCB would allow (1) an untrusted database application or an unprivileged database user to access operating system objects in an unauthorized manner, or (2) an untrusted operating system application or an unprivileged operating system user to access database objects in an unauthorized manner. Furthermore, when non-discretionary access controls are implemented in both the operating system and the database subsystem, the composition of the two should not introduce covert channels that were not present when the individual policies were supported.

The suggested composition causes the access control partitioning of the TCB into an operating system and a database partition. The two partitions can share other TCB policy families such as identification and authentication, system entry, and trusted channel. Other similar examples of policy partitioning are offered by message or mail subsystems and communication protocol subsystems.

Composition by Policy Subsetting. An alternate method of policy composition is that provided by policy subsetting. In this method, separate TCB subsets are allocated different policies. This method of policy composition is addressed in detail in the *Trusted Database Interpretation (TDI).*[2]

In this composition method a TCB subset, M, is a set of software, firmware, and hardware (where any of these three could be absent) that mediates the access of a set of subjects, S, to a set of objects, O, on the basis of a stated access control policy, P, and satisfies the properties or the reference validation mechanism.[3] M uses resources provided by an explicit set of more primitive TCB subsets to create the objects of O, create and manage its data structures, and enforce the policy P. (The above definition does not explicitly prohibit an access control policy P that allows trusted subjects.) If there are no TCB subsets more primitive than M, then M uses only hardware resources to instantiate its objects, create and manage its own data structures, and enforce its policy. However, if M is not the most primitive TCB subset, then M does not necessarily use the hardware or firmware functions to protect itself. Rather, it uses either hardware resources or the resources provided by other, more primitive TCB subsets. Thus TCB subsets build on abstract machines, either physical hardware machines or other TCB subsets. Just like reference validation mechanisms, a

---

[2] U. S. Department of Defense, National Computer Security Center (NCSC). April 1991. *Trusted Database Management Systems Interpretation of the Trusted Computer Systems Evaluation Criteria, Version 1.* NCSC-TG-21. Fort George G. Meade, MD: NCSC.

[3] Ibid.

TCB subset must enforce a defined access control policy separately from those policies enforced by other subsets.

The access control policy P[i] is the policy allocation for each identified TCB subset M[i] of a product along with the relation of these policies to the product policy P. The allocated policies P[i] will be expressed in terms of subjects in S[i] and objects in O[i]. To satisfy the requirement that the (composite) TCB enforce its stated policy P, each rule in P must be traceable through the structure of the candidate TCB subsets to the TCB subset(s) where that enforcement occurs. It must also be noted that every subject trusted with respect to P[i] must be within the TCB subset M[i].

APPENDIX C. NON-REPUDIATION[1]

C.1 NON-REPUDIATION CRITERIA

Repudiation is denial by one of the entities involved in a communication of having participated in all or part of the communication. Non-repudiation is a security service that protects one participant in a communication from a false denial of participation by another participant. Several varieties of non-repudiation have been defined for different types of applications, e.g., non-repudiation with proof of submission, proof of delivery, and proof of receipt are defined for messaging applications. However, all of these can be reduced to special cases of the basic non-repudiation service: non-repudiation with proof of origin. The criteria defined below are stated in terms of the basic non-repudiation with proof of origin service.

Provision of non-repudiation services ultimately requires an extensive set of procedural security measures that lie outside the scope of traditional evaluation criteria for a computer system, much less a host TCB. Thus the criteria articulated below describe requirements that provide a technical foundation for non-repudiation in a distributed system, but these requirements must be combined with procedural security measures to provide a complete set of requirements for non-repudiation services.

If a distributed system supports non-repudiation, it shall incorporate facilities for protecting transmitted, application layer data objects, i.e., messages, in support of this security service. (The term "messages" is used here to refer to such objects, but this term should not be interpreted narrowly; a message in this context is equivalent to a document and is potentially a multi-media object.) The following requirements must be satisfied in order for a distributed system to offer the non-repudiation service:

a. The system shall incorporate facilities to ensure the integrity of a message for which non-repudiation is offered.

---

[1] This appendix was provided via electronic mail by Stephen Kent of Bolt Beranek and Newman, Inc. Minor changes by IDA have been incorporated. Significant modifications by IDA to the original are indicated by brackets ("[ ]").

b. The system shall be capable of identifying the originator of the message, and shall bind that identity to the message.

c. The system shall provide facilities to associate with the originator any contextual information that is necessary for correct interpretation of a non-repudiable message sent by that originator.

d. The system shall enable its users to establish the existence of a message at a point in time, through the use of an independent time and date source.

e. The system shall provide facilities that support declaration of the semantic context of a message accorded non-repudiation.

f. The system shall include provisions for resolving repudiation disputes through the use of an independent authority.

g. The system shall provide a trusted facility by which a user invokes application of non-repudiation services.

h. The system shall provide a facility to display a message and all relevant information supportive of non-repudiation. This information includes the identity of the originator and, if required, the identity of recipients, the basis for validating these identities, any per-originator contextual information associated with the message, time context information associated with the message, semantic context information, and dispute resolution context information.

Subsequent sections describe the criteria for these requirements in more detail.

C.1.1 Message Integrity

A message accorded non-repudiation shall be protected against attempts by the originator, recipients, or third parties, to modify the message in an undetectable fashion, or to produce another message with different semantics but with syntax that is indistinguishable by the mechanisms that implement the non-repudiation services.

If message integrity for non-repudiation is brought about by means that securely archive the entire message, then this requirement may be trivially met. If some form of message integrity function is employed to represent the whole message, then this function shall exhibit the properties of a strong one-way hash function (e.g., the Secure Hash Standard[2]).

---

[2] [U. S. Department of Commerce, National Institute of Science and Technology (NIST). April 1993. *Secure Hash Standard.* FIPS PUB 180.]

C.1.2 Originator Identification

The message originator shall be identified in an unambiguous fashion, relative to the context in which the communication takes place and in which disputes will be resolved. If the semantic context for the message requires that recipients be identified, then the same requirements apply to the recipient identifiers. An identifier employed for non-repudiation may refer either to a named individual, or to an organizational role occupied by an individual.

If the originator is identified in terms of an organizational role, then there is no requirement for an externally (outside the organization) visible, individual identifier. However, to support individual accountability internal to a system, an organization shall identify the individual acting in a role at any point in time.

The identities employed for non-repudiation shall be mapped to "real-world" names for these parties to the communication. This mapping shall be brought about via a mechanism that is trusted by all of the parties to the communication and by any entity that may be called upon to resolve disputes. For example, if an electronic mail address or a login name is used to identify the originator of a message for non-repudiation purposes, then the mapping between this identifier and the individual or role it represents shall be at least as trustworthy as the mechanism used to bind the identifier to the message for non-repudiation purposes. This mapping shall be provided by a mechanism that is visible and auditable by recipients and by parties responsible for dispute resolution. If distinguished names (as per X.500[3]) are employed as identifiers, these may provide the requisite degree of real-world descriptiveness without need for an additional layer of mapping.

C.1.3 Originator Contextual Information

In some applications, per-originator contextual information is required by recipients to interpret a message. A system supporting such applications shall provide a facility for associating, with the originator, any contextual information pertinent to interpreting the originator's signature. If non-repudiation support is offered only within an application, then per-originator contextual information may be tailored to the application. If the distributed system provides generic support for non-repudiation, then support for per-originator contextual information shall be generic and extensible. For example, a generic system should

---

[3] [International Telegraph and Telephone Consultative Committee (CCITT). 1988. *Information Technology—Open Systems Interconnection—The Directory: Overview of Concepts, Models, and Sevices.* Recommendation X.500. Geneva, CH: CCITT.]

be able to express the notion that some originators may be authorized to commit an organization to multi-million dollar contracts, others may be limited to authorizing purchases under $10,000, and others may have no fiduciary authority at all.

If the role in which an originator acts is explicitly part of the originator's identification, it may provide sufficient context for purposes of dispute resolution for many applications. In such case, no additional contextual information need be provided by a system in support of non-repudiation. For example, a message originated by someone acting in the role of president of a company would generally be construed to be sufficient for a wide range of fiduciary authorizations.

When per-originator contextual information is required, it shall be accessible by the recipients of the message and by a dispute resolution authority, as this information represents an important part of the semantic context for interpreting a message. If public-key digital signature facilities are employed, originator contextual information may be incorporated into a digitally signed "context" certificate, analogous to public-key identification certificate. A context certificate may be bound to the user through the identity contained in an originator's identification certificate.

As an alternative to certificate-based solutions, originator contextual information may be specified indirectly by reference based on the originator identifier and registered on a per-organization, per-originator, or bi-lateral basis. If this latter approach is employed, a secure binding shall be maintained between the originator identification information and the contextual information. This latter form of binding may be implemented via technical or procedural means.

C.1.4 Digital Signature Mechanisms

Means of generating digital signatures using either symmetric (secret key) or asymmetric (public-key) cryptosystems are defined in the literature (Rabin, Smid, RSA, ElGamal),[4] and either type of signature technology may employed for non-repudiation purposes. A digital signature used for non-repudiation and based on symmetric cryptography shall employ a notarization facility. A signature used for non-repudiation and based on asymmetric cryptography shall make use of public-key certificates. When a symmetric, notarized signature is employed, the notary also may provide a timestamping service (see [C.1.7]) concurrent with the signature service.

---

[4] [No citations given in original.]

The key used to cryptographically protect the hash value shall be made available only to the entity that is identified as responsible for that key (or to a notarization facility in the case of signatures based on symmetric algorithms). A cryptographic module shall be employed to protect this key and to perform the signature operation. The module shall conform to FIPS PUB 140-1 specifications. [The module shall also conform to the requirements of the Cryptographic Domain Protection and Secure Key Management families of the Cryptographic Support class.]

C.1.5 Signature Credentials

If a notary-based, symmetric signature system is employed, a secret key representing the originator is registered with the notary who shall establish and maintain the correspondence between this key and the identity that it represents. The notary also may maintain the binding to any contextual information associated with the originator. The notary shall make available the identity of the originator (and contextual information) associated with a signature to any (authorized) recipient and to any third party called upon to resolve a non-repudiation dispute. The techniques used by the notary to represent signature credentials must ensure the integrity and authenticity of the communication between the notary and the recipients or dispute resolution parties.

If public-key certificates are employed, the private key associated with an originator should be available only to the originator, and the identity binding is provided through a public-key certificate. This certificate shall be signed by a party (a Certification Authority) trusted by the originator, recipients, and dispute resolution entities to correctly identify the originator (and to assume liability for incorrect identification). Identity certificates may take on various forms, but use of the international standard X.509[5] (1993 and later) is recommended in support of interoperability.

If public-key certificates are employed and are organized in a hierarchic fashion, the requirements cited here apply to each tier in the hierarchy. The recursion implied by hierarchic organization shall be terminated by an out-of-band mechanism that allows recipients to validate signature credentials for a root of the hierarchy.

---

[5] [International Organization for Standardization (ISO). 1988 (revised 1995). *Information Technology—Open Systems Interconnection—The Directory: Authentication Framework*. IS088 ISO/IEC 9594-8. Recommendation X.509. Geneva, CH: ISO.]

C.1.6 Revocation of Signature Credentials

If a distributed system providing non-repudiation makes use of digital signature credentials, such as public-key certificates, then the system shall incorporate a facility to revoke these credentials. Signature credentials shall be revoked when there is a compromise of private keying material or when the identity or contextual information bound to a signature key is no longer valid.

The issuer of signature credentials shall provide a means by which the act of revoking these credentials is made known to potential recipients and to third parties used for dispute resolution. An issuer of signature credentials shall provide either a real-time capability for determining the current validity of a specified originator's credentials, or shall provide a periodic declaration of revoked credentials.

If a real-time inquiry facility is provided, it may operate in one of two ways. The response to an inquiry may itself be a non-repudiable message traceable to the issuer. Alternatively, the response may be an authentic and integrity-checked message that reports the state of a trusted archive that records the status of all credentials signed by the issuer (for a period of time specified by the issuer).

If the issuer provides a periodic declaration of revoked certificates, then this declaration shall take the form of a non-repudiable message traceable to the issuer. Posting a list of revoked certificates to a directory in accordance with the Certificate Revocation List (CRL) format defined in X.509[6] (1993 and later) satisfies this later requirement.

Computers via which the status of signature credentials can be ascertained, either via real-time inquiry or via posted CRLs, shall be accessible to all recipients and dispute resolution parties associated with all originators served by the issuer. These computers also must exhibit an availability consistent with the demands of the applications that make use of non-repudiation functions.

There is no uniform requirement for the timeliness with which the system must make the revocation information available for recipients, but it must always be possible for a recipient to establish an upper bound for how long it will take to acquire the requisite supporting information (exclusive of delays associated with system availability).

If signature credentials are managed in a form that involves a hierarchy of issuers, then the credential revocation requirements apply at each tier of the hierarchy. There must

---

[6] [Ibid.]

be an out-of-band termination for this recursion, to address revocation of root credentials. For example, a trusted archive could be used to record revocation actions for the root(s) of the hierarchy. Different tiers may elect to use different means of complying with these requirements. For example, some tiers may employ real-time attestations of credential validity whereas others may issue CRLs.

C.1.7 Message Time Context

A system supporting non-repudiation shall provide a facility that enables an originator or a recipient to establish the existence of a message, in its possession, at a fixed point in time. This facility shall associate with the message a time (and date) stamp. The timestamp must be from a source considered trustworthy by the originator, recipient, and by any party called upon for dispute resolution. This criteria establishes no specific requirements to the accuracy or precision of the timestamp, as these requirements may be application specific. This facility may be implemented either through trusted archiving or by issuing a digitally signed message that binds a timestamp to the original message.

If archiving is employed, the message (or message hash) shall be recorded in a serialized, timestamped fashion by trusted third-party. The archival function may be provided by the party that applies the timestamp, or it may be provided by a separate, perhaps distributed, entity that retains and/or publishes timestamped, hierarchic hashes from other sources (cf., patented Bell Labs approach).

If the timestamp entity issues a digitally signed message attesting to the existence of another message at the indicated point in time, then responsibility for retaining this timestamp message lies with the recipient of this timestamp message, rather than with some trusted archive entity. The signature applied by the timestamp entity is itself subject to repudiation, and thus non-repudiation measures must be applied to these signed timestamp messages as well. The recursion implied by this procedure must terminate via some out-of-band means.

C.1.8 Message Semantic Context

A message suitable for non-repudiation must be unambiguous within some established, semantic context. For example, a message text that states only "OK, Bill, you have permission to proceed" cannot be made non-repudiable by digitally signing it and affixing a timestamp. The inherent ambiguity of the message precludes it as a candidate for non-repudiation. If a system provides non-repudiation services, each application making use of such services must provide facilities to support this requirement in the context of that appli-

cation. For example, the use of highly constrained electronic data interchange (EDI) message formats is one way to establish semantic context for a class of message exchanges.

C.1.9 Dispute Resolution Context

Non-repudiation is predicated on the existence of an independent third party empowered to resolve disputes regarding (allegedly) repudiated messages. The third party must be trusted by the originator and by recipients to impartially resolve disputes and these parties must agree, in advance, to employ a specific dispute resolution procedure. This advance agreement is required to ensure that the signature algorithms, credentials, revocation, and archiving mechanisms are all consistent with the requirements imposed by this third party.

If the dispute resolution context is defined on an per-application basis, then no explicit declaration of the context need be provided in messages, signature credentials, etc. Instead, implicit means may be employed to identify the dispute resolution context.

A distributed system that provides general non-repudiation support must provide explicit means of identifying the dispute resolution context employed for each application. This identification shall include the technical procedures and the class of arbiters to be employed in case of a dispute. The identification shall be brought about through the use of unique identifiers, e.g., ISO Object Identifiers (OIDs), and a registration infrastructure that maps these identifiers to registered contexts.

C.1.10 Control and Audit of Non-Repudiation Services

A system that offers non-repudiation services must provide positive control over the invocation of these services by the originator of a message. For example, prior to affixing a digital signature to a message for non-repudiation purposes, the user might be required to acknowledge this action explicitly.

If a system provides a trusted path facility, this facility shall be used to control invocation of non-repudiation services.

The system shall incorporate a means of auditing the invocation of a non-repudiation service by a message originator.

C.1.11 Display of Non-Repudiation Information

A system supporting non-repudiation shall include facilities to provide a trusted display of a message and all ancillary information necessary for support of non-repudiation.

This display facility shall be available to the message originator, recipients, and to third parties called upon for dispute resolution.

The ancillary information includes the identity of the originator and, if required, the identity of recipients, as well as the basis for validating these identities. For example, in a system that makes use of public-key certificates, a facility must be provided to display the chain of certificates needed to validate these identities and the associated certification revocation status information.

The system must provide a means for displaying any per-originator contextual information, time context information, explicit semantic context information, and explicit dispute resolution context information. If multiple timestamps are associated with a message, then the system must be capable of displaying each timestamp, the identity of the timestamp authority, etc.

If a system provides a trusted path facility, this facility shall be used for display of messages and ancillary non-repudiation information.

C.2 NON-REPUDIATION CRITERIA RATIONALE

Repudiation of a message can adversely affect the recipients who acted upon the message in good faith. An originator could repudiate a message by in various ways:

a. He can deny that he ever sent the message in question, e.g., that the signature is invalid or that the content is not what he signed or what he intended to sign.

b. He can deny that he sent the message at the time in question.

c. He can argue that the recipient's interpretation of the message is erroneous, e.g., that the message is being interpreted in a different context than was intended.

d. He can claim that the technical means by which his digital signature was affixed to the message was compromised, and that someone else affixed his signature.

One can generate a complementary set of repudiation concerns associated with a recipient by transforming each of the ones noted above. For example, a recipient can deny having received a message or can deny receiving it at a specific point in time. Thus, in the context of a specific transaction, both originator and recipient repudiation concerns must be addressed.

Some forms of repudiation are partially countered by purely technical means at the originator and destination, through the use of digital signatures. Such signatures prevent (undetected) modification of signed messages. Some require the use of a trusted third party to affix a timestamp to a message. This is necessary to counter attempts to repudiate messages through intentional loss or compromise of private keys, or arguments based on the time context in which a message was sent or received. Some repudiation attacks must be countered through the [means used] to specify the semantic context for message interpretation.

Provision of non-repudiation services ultimately requires an extensive set of procedural security measures that lie outside the scope of traditional evaluation criteria for a computer system. Thus the criteria articulated in [C.1] describe requirements that provide a technical foundation for non-repudiation in a distributed system, but they must be combined with procedural security measures to form a complete set of requirements for non-repudiation services.

Derived forms of non-repudiation service can all be modeled using non-repudiation with proof of origin. This simplification is brought about by applying the labels of "originator" and "recipient" to appropriate entities in the communication path and by defining what constitutes a "message" in a context sensitive fashion.

For example, the non-repudiation with proof of submission service defined in X.411[7] can be viewed as non-repudiation with proof of origin by considering the source message transfer agent (MTA) to be the originator of a proof of submission message directed to the "real" message originator who now becomes a recipient. Non-repudiation with proof of delivery involves a destination MTA acting as an originator of a "proof of delivery" message directed to the originator of the message in question. Likewise, non-repudiation with proof of receipt can be offered by having the recipient of the message in question act as an originator and transmit a "proof of receipt message" to the originator of the original message who now is viewed as a recipient.

Subsequent sections describe the rationale for the criteria given in [C.1.1] through [C.1.11].

---

[7] [International Telegraph and Telephone Consultative Committee (CCITT). 1988. *Message Handling Systems—Message Transfer System: Abstract Service Definition and Procedures*. Recommendation X.411. Geneva, CH: CCITT.]

C.2.1 Message Integrity Rationale

One might imagine non-repudiation built on some form of trusted archive system, in which case message integrity might be ensured by retaining a complete copy of the message. This would eliminate concerns about possible tampering with a message by originators or recipients, because the "reference" copy of the message would be maintained by the trusted (third-party) archive. In fact, the Latin Notaire model used in common law countries is a physical realization of this approach. However, such a system could create enormous storage demands, and thus most designs for (electronic) non-repudiation are based on the use of digitally signed message integrity functions.

The integrity requirement here is much more stringent than that usually applied to two-party communications where protection is required against attacks by a third party that is not part of the communication. In two-party integrity, simpler forms of integrity mechanisms can be employed (e.g., Data Encryption Standard (DES)[8] [Medium Access Control] MAC), whereas for non-repudiation one requires strong, one-way hash functions. The integrity requirement for non-repudiation is more analogous to, but not equivalent to, the problem of authenticating multicast messages. The critical difference in the non-repudiation context is that the originator, not just the recipients, might attempt to fabricate alternative messages with different semantics that will be interpreted as valid by one or more recipients.

C.2.2 Originator Identification Rationale

The requirement for mapping to "real-world" names is based on the assumption that resolution of non-repudiation disputes may ultimately require legal proceedings. In a court of law, it would be necessary to establish the correspondence between some name form used in the electronic context, and names used in a civil or common law context. This correspondence, if not explicit and obvious from signature credentials, would require an ancillary, secure translation. This translation must then be as secure as the credential issuance and binding procedures, or it could be used to undermine the non-repudiation service. The principle of least privilege argues in favor of eliminating additional name translation steps thorough the use of highly descriptive names.

---

[8] [U. S. Department of Commerce, National Bureau of Standards. January 1977. *Data Encryption Standard.* FIPS PUB 46. Gaithersburg, MD: National Institute of Science and Technology.]

C.2.3 Originator Contextual Information Rationale

Some of the contextual information associated with a message will come from the message itself. For example, an EDI-formatted purchase requisition conveys substantial context through its format. However, the originator of a message must be "appropriate" if the message is to be acted upon by recipients. For example, a well-formed EDI message carries no import if it is signed by an employee who has no authority to place orders. Thus there often is a need for originator-specific contextual information to complete the semantic context for interpretation of signed messages.

Usually, the originator-specific information is a form of authorization, e.g., a fiduciary authorization. This information can be maintained separately from identity credentials, and can be bound to identity through external, perhaps procedural, means. However, the means by which this binding is brought about must be well understood by the originator, recipients, and by dispute resolution arbiters. The binding between the signer's identity and this other information also must be as secure as the rest of the signature facility, or it may become the weak link in the non-repudiation chain.

C.2.4 Digital Signature Mechanisms Rationale

Digital signatures based on symmetric cryptography are generally inefficient and unwieldy (cf., Rabin scheme) unless a third-party notarization function is employed (NIST special pub by Smid).[9] Techniques not based on the use of notaries generally are not considered practical, especially in light of the availability of public-key approaches. Hence the decision to require use of a notary function for any symmetric cryptographic signature mechanisms when used for non-repudiation purposes.

For both symmetric and asymmetric signature approaches, the top-level procedures for generating and validating a digital signature are analogous. An originator generates a digital signature for a message by applying a one-way hash function to the message. Then the value is cryptographically sealed using a key associated with the originator. The signature is validated by calculating the one-way hash value of the received message, and performing a computation using this locally computed hash value, the received (sealed) hash value, and a key associated with the sender. The details of signature generation and validation differ, based on the specific signature algorithm employed.

---

[9] [No citations given in original.]

It is essential to the concept of non-repudiation that the key used to sign a message be protected against disclosure or unauthorized use. This goal must be reconciled with the need for high quality key generation. Ideally, a private signature key is generated using a hardware random number generator in a tamper-resistant, personal cryptographic token. In this fashion the key need never be available outside of the token in plaintext form. Other approaches make use of keys generated by a certification authority (CA) and provided to a user for use in a token. However, such approaches raise the specter of unauthorized use of a private key by the CA or disclosure of a private key in transit between the CA and a user.

Protection of a signature private key using only software is problematic. Exposure of the key during the signing operation is unavoidable. It may be mitigated by the assurance level of the TCB, assuming good, modular design practice, but the procedural and physical security of the computer on which the signing takes place also must be considered. There is often a strong desire for fine-grained use of private signature keys, for accountability, and for portability of private keys, to accommodate individual mobility. Thus the computers on which signatures may be implemented in software might typically be desktop, laptop, and notebook machines. In such circumstances, the overall security of such computers is general well below what can be accomplished using hardware tokens such as smart cards and SmartDisks.

For most individuals or roles, loss of use of a private signature key is not critical, in that issuance of a new public key in a new certificate can restore signature operation. Long-term validation of signed messages is not dependent on access to private signature keys. (This in contrast to symmetric schemes using a notary, where long-term availability is a requirement.) Since signature keys need not be used for encryption key management, there is no implied loss of data associated with loss (not disclosure) or a signature private key. Thus concerns for backup or archival retention of private signature keys ought not result in exposure of these keys to other parties. Discussions about key "escrow" procedures, whether by government or commercial escrow agents, need not apply to signature private keys, in so far as these keys are not used to conceal information.

C.2.5 Signature Credentials Rationale

In a notary-based signature system, the users trust the notary to protect user keys and to validate signatures correctly. A notary can forge messages under the identity of any of the users it serves. These messages are indistinguishable from messages signed by the user.

In a public-key certificate system, the private key is available only to the originator and the identity binding is provided through a public-key certificate signed by a CA. Here the CA need not be able forge messages through knowledge of the user's private key (if the key is not disclosed to the CA). However, a CA can sign a certificate with the user's identity but containing a different public key, one for which it has access to the private component. In this circumstance, the CA can create signed messages that appear to originate from the user, although they will differ in the public key needed to validate the signature. To repudiate such messages, the user would have to demonstrate that the certificate in question was not issued in response to a request from that user, e.g., by demanding that the CA produce documentation of the user's submission of a public key to be certified. Thus a claim of forgery by a CA is more readily resolved, assuming good procedural security practices are in place for user registration.

A certificate is a means of leveraging the effort required to securely distribute a public key, i.e., distributing the public key of a CA has the effect of distributing the public keys of all the users served by the CA. A hierarchy of CAs is a recursive application of this sort of leveraging. Most proposals for large certification systems serving diverse user communities are based on use of a hierarchic CA system. Such systems may contain a single root, or may contain multiple roots that "cross-certify" one another, in support of cross-domain interoperability. Examples include the Internet certification system (RFC 1422)[10] and the NIST Public Key Infrastructure.[11]

C.2.6 Revocation of Signature Credentials Rationale

Revocation is a natural by-product of the use of signature credentials. Such credentials are a form of capabilities and thus there must be a means to revoke them in response to compromises, identity changes, and authorization changes. Use of hardware tokens for key storage and signing can minimize compromises, but identity and authority changes are inevitable in most systems.

Real-time attestation to credential validity is potentially expensive in terms of communication and computational loads, especially for large user communities. It also may imply increased exposure for private keying material of the issuer and require polyinstantiation of such material to keep up with transaction processing demands. Thus this approach

---

[10] [Internet Engineering Task Force (IETF). February 1993. *Request For Comment (RFC) 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management.]*

[11] [No citations given in original.]

to revocation should be employed only when necessitated by application requirements for very timely revocation.

The CRL model of revocation scales well by allowing ready distribution of the communication and processing load associated with certificate status queries. A credential issuer need not engage in real-time signature computation to support the CRL model, which also bodes well for the security that can be afforded to the issuer's private keying material. However, the CRL approach is not highly responsive and thus may not meet the requirements of some applications. Careful choices for credential validity intervals can help reduce the size of CRLs, but the costs of reissuing credentials argue against very short validity intervals.

A very different approach to revocation management is based on trusted archiving of revoked certificate lists (or of valid certificates). The archive is trusted to retain lists of revoked (or valid) certificates (or their unique identifiers) for a long period of time. These lists need not be signed, but the archive must be trusted to retain the lists accurately, to associate the certificates with the proper issuers, and to accurately record when revoked certificates were registered in the archive. In this approach, recipients and dispute resolution authorities need not collect and retain CRLs, but need merely query the archive to determine if a given credential has been revoked and, if so, when. The use of a trusted archive facility terminates the recursion implied by digitally signed CRLs. Thus trusted archiving for revocation of issuer signature credentials, especially at higher tiers of a hierarchy, is especially attractive. A trusted archive might grow unmanageably large if it were required to retain a very large number of revoked (or valid) credential identifiers for a very long time. Yet the lifetime of CRLs might easily be seven years or more for many financial transactions. However, signature credential issuers generally may be expected to have a lower rate of revocation and a longer credential lifetimes than users, making this approach more feasible.

The recipient of a signed message must wait some amount of time to acquire revocation data in support of non-repudiation. Even if a real-time certificate query system is employed, there will be a delay between the time a compromise occurs, it is detected, and it is reported to the CA. Thus instantaneous verification of the status of a certificate, with regard to compromise, is not technically practical. A prudent recipient may elect to postpone performing some action, in response to receipt of the signed message, until the requisite revocation data becomes available. Thus it is critical that the recipient know how long he must wait before the requisite data will be available.

C.2.7 Message Time Context Rationale

The time at which a message existed in the possession of an originator or recipient is a critical element of non-repudiation. If a recipient can demonstrate that he was in possession of a message at a specified time, then the originator must have sent the message prior to that point in time. For example, by assigning the role of originator to message relay agents, the use of timestamps provides a basis for proof of submission and proof of delivery that non-repudiation guarantees.

In general, inclusion of a timestamp by an originator of a message, e.g., as part of the message body, is not a good source of time for non-repudiation purposes because the originator is not considered to be trusted. However, inclusion of such a value can indicate a declaration by the originator of the time at which a message was prepared or transmitted and that may contribute to the semantic context of the message, assuming that timestamps applied by other parties are consistent with the one provider by the originator.

The requirements for precision and accuracy of timestamps may vary considerably among applications, hence the lack of a general requirement in the criteria. For some applications, mere serialization of events will suffice, whereas other applications may require sub-second accuracy. With the advent of inexpensive Global Positioning System (GPS) receivers and protocols such as Network Time Protocol (NTP) III, very precise, very accurate synchronized time is widely available in worldwide networks, making it possible to apply timestamps that exhibit high precision and accuracy.

Since the digitally signed timestamps bound to message hashes are themselves messages, the possibility of repudiation of these messages arises. This argues for very high assurance implementation of timestamping facilities, especially with regard to protection of private keys used to sign the messages. In a hierarchic certificate system, this also argues for the timestamp server to have a certificate issued by a CA very near the root, if not the root itself.

An alternative to digitally signed timestamps is a hash tree approach developed and patented by scientists at Bell Labs. This technique constructs a binary tree of one-way hash values based on messages submitted by users. Periodically, the root of the tree is computed and the result is published in multiple locations, e.g., in the *New York Times*, making tampering with the root value impractical. Each user then receives a sequence of hash values that represent the path from their submitted hash to the root. A user can check the path against the published root to ensure that the path is suitable for later use in support of a non-

repudiation claim based on the time interval in which the user's hash was submitted. No secret quantities are involved, either at the server or the users. All messages received with one time interval are equivalent in terms of timestamping, which argues for a shorter interval, but shorter intervals generate more roots that must be published and distributed. More work is needed to determine what intervals are most appropriate.

Another approach to timestamping is a trusted archive that timestamps and serializes all entries, where the entries may be messages or the hash values thereof. This approach leads to a rather substantial storage requirement, even if only hash values are stored, as well as a very stringent trust requirement for the server, and thus does not seem especially attractive for large communities.

C.2.8 Message Semantic Context Rationale

To benefit from non-repudiation services, a message must have a well-defined, unambiguous semantics. Otherwise, legitimate differences in the interpretation of the message, by the originator, recipients, and dispute resolution parties, would negate the non-repudiation technology. This is an application-specific requirement that cannot be addressed by generic TCB or system security measures. Despite this constraint, there are still technical means that can be used to minimize ambiguity in many instances. For example, use of rigid, application-specific formats, such as EDI message formats, can significantly reduce ambiguity in a standard way. In contrast, interpersonal messages using natural language require careful composition to avoid the sorts of ambiguity that could negate non-repudiation services.

C.2.9 Dispute Resolution Context Rationale

Specification of the dispute resolution procedure is an essential prerequisite to effective non-repudiation, in a procedural sense. If two or more parties enter into communication using non-repudiation techniques, but fails to agree upon a dispute resolution procedure and or mutually agreeable arbiter, then a dispute may be unresolvable.

Over time there may not be a need to explicitly declare the procedure and arbiter for each message. Instead, as disputes arise in certain application contexts, they may come to be arbitrated under a common set of ground rules and the set of arbiters may also be defaulted. However, in a more abstract sense, this merely constitutes an implicit declaration of the dispute resolution context based on the application context. Moreover, in international transactions, there may be a need for more explicit specification of the dispute resolution context due to widely varying requirements under different national laws.

Differences in the technical requirements imposed by different arbiters are easy to imagine. For example, one third party might require that it act as a timestamp notary, and archive the message hashes that it will later be called upon to evaluate. Another third party might require the receiver to acquire and retain the sender signature credentials, CRLs, and timestamps for messages that the receiver considers candidates for repudiation. Thus it is critical that the requirements for dispute resolution be established prior to engaging in communication requiring non-repudiation services.

C.2.10 Control of Non-Repudiation Services Rationale

The action of processing a message for non-repudiation purposes, e.g., affixing a digital signature to a message, must be carefully controlled. Otherwise, malicious software could misapply non-repudiation mechanisms to data that a user does not wish to so process. For example, a message constructed by an attacker and infiltrated into a user's system might be signed on behalf of the user, resulting in a later repudiation attempt by the user. To ensure that non-repudiation mechanisms are applied only when the user wishes to apply them, the system must not only enforce access control, to prevent invocation by other users, but also must employ mechanisms to prevent unintended invocation.

Use of mandatory (rule-based) access controls is one approach to avoiding the threat of unauthorized invocation by malicious software, but one must be careful with such approaches. For example, a special sensitivity (e.g., "system low") might be established for data that are to be signed, and an explicit downgrade operation might be required to enable the data to be signed. However, this approach conflicts with legitimate use of sensitivity markings for data, where data to be signed might require a high marking.

The fundamental requirement here is that of a trusted path, so that the user is assured that the data he is about to sign is that which he intends, and that it is the user, not some malicious software, that is requesting the non-repudiation operation. In (lower assurance or less security functional) systems that do not support trusted path, or where the trusted path facility cannot accommodate display of messages, alternative mechanisms must be employed in an effort to achieve the same effect.

C.2.11 Display of Non-Repudiation Information Rationale

It is critical that an originator be able to display the content of a message to which he is about apply non-repudiation techniques. Otherwise, the originator might sign a message that is other than what he imagined. In building a high assurance non-repudiation system, it may be much easier to provide trusted display applications that provide trusted

message creation applications. Thus the criteria place an emphasis on the ability to display the message and ancillary data, rather than on the ability to construct the message securely.

In systems that provide a trusted path between the user and the TCB, high assurance display and control of non-repudiation should make use of the trusted path mechanism.

A message to which non-repudiation services are applied begins without a timestamp, but acquires one or more timestamps as it is processed. Thus, it is important to be able to display each timestamp.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1995	Final

4. TITLE AND SUBTITLE
Security Criteria for Distributed Systems: Functional Requirements

5. FUNDING NUMBERS
DASW01-94-C-0054

Task Order T-AA5-962

6. AUTHOR(S)
Terry Mayfield, Virgil D. Gligor, Janet A. Cugini, John M. Boone, Robert W. Dobry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-3159

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Security Agency
Attn.: C41
9800 Savage Rd.
Ft. George G. Meade, MD 20755-6000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; unlimited distribution.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

This document specifies functional security requirements that can be used to assemble evaluation-oriented criteria for distributed computer systems. These requirements capture necessary security characteristics of distributed systems, enable the definition of specific protection profiles (evaluation criteria) for trusted distributed systems that can be used in various threat environments, and allow for protection profile extension and refinement which may be needed as technology evolves, threats change, and experience is gained in specifying and evaluating distributed systems. These requirements are presented in a modularized format in order to avoid some of the extensibility drawbacks of earlier computer security criteria. Requirements are specified for 12 computer security functional classes: Trusted Computing Base, Identification and Authentication, System Entry, Trusted Path, Data Confidentiality, Data Integrity, Cryptographic Support, Access Control, Covert Channel Countermeasures, Audit, Availability, and Security Management.

14. SUBJECT TERMS
Computer Security, Distributed Systems, Trusted Computing Base, Federal Criteria, Common Criteria.

15. NUMBER OF PAGES
320

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR